



Architecture d'une coopération de gestions de fichiers dans le cadre d'un réseau local hétérogène

Merval Jurema, Amadou Niang

► To cite this version:

Merval Jurema, Amadou Niang. Architecture d'une coopération de gestions de fichiers dans le cadre d'un réseau local hétérogène. Génie logiciel [cs.SE]. Ecole Nationale Supérieure des Mines de Saint-Etienne; Institut National Polytechnique de Grenoble - INPG, 1981. Français. NNT : . tel-00811869

HAL Id: tel-00811869

<https://theses.hal.science/tel-00811869>

Submitted on 11 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre 22 II

THÈSE

présentée par

Merval JUREMA

Amadou NIANG

pour obtenir

LE TITRE DE DOCTEUR-INGÉNIEUR

"SYSTEMES ET RESEAUX INFORMATIQUES"

ARCHITECTURE D'UNE COOPERATION DE GESTIONS DE FICHIERS DANS LE CADRE D'UN RESEAU LOCAL HETEROGENE

Soutenue à Saint-Etienne le 29 Juin 1981 devant la commission d'examen :

MM. L.BOLLIET

Président

J.F. CHAMBON

J.C. CHUPIN

S. GUIBOUD-RIBAUD

} Examineurs

N° d'ordre 22 II

THÈSE

présentée par

Merval JUREMA

Amadou NIANG

pour obtenir

LE TITRE DE DOCTEUR-INGÉNIEUR

"SYSTEMES ET RESEAUX INFORMATIQUES"

ARCHITECTURE D'UNE COOPERATION DE GESTIONS DE FICHIERS DANS LE CADRE D'UN RESEAU LOCAL HETEROGENE

Soutenue à Saint-Etienne le 29 Juin 1981 devant la commission d'examen :

MM. L.BOLLIET

Président

J.F. CHAMBON

J.C. CHUPIN

S. GUIBOUD-RIBAUD

} Examineurs

Président : M. Philippe TRAYNARD
Vice-Présidents : M. Georges LESPINARD
M. René PAUTHENET

PROFESSEURS DES UNIVERSITES

MM ANCEAU François	Informatique fondamentale et appliquée
BESSON Jean	Chimie Minérale
BLIMAN Samuël	Electronique
BLOCH Daniel	Physique du Solide - Cristallographie
BOIS Philippe	Mécanique
BONNETAIN Lucien	Génie Chimique
BONNIER Etienne	Métallurgie
BOUVARD Maurice	Génie Mécanique
BRISSENEAU Pierre	Physique des Matériaux
BUYLE-BODIN Maurice	Electronique
CHARTIER Germain	Electronique
CHENEVIER Pierre	Electronique
CHERADAME Hervé	Chimie Physique Macromoléculaire
MmeCHERUY Arlette	Automatique
MM CHIAVERINA Jean	Biologie, biochimie, agronomie
COHEN Joseph	Electronique
COUMES André	Electronique
DURAND Francis	Métallurgie
DURAND Jean-Louis	Physique Nucléaire et Corpusculaire
FELICI Noël	Electrotechnique
FOULARD Claude	Automatique
GUYOT Pierre	Métallurgie Physique
IVANES Marcel	Electrotechnique
JOUBERT Jean-Claude	Physique du Solide - Cristallographie
MmeJOURDAIN Geneviève	Traitement du Signal
MM LACOUME Jean-Louis	Géophysique - Traitement du Signal
LANCIA Roland	Electronique - Automatique
LESIEUR Marcel	Mécanique
LESPINARD Georges	Mécanique
LONGEQUEUE Jean-Pierre	Physique Nucléaire Corpusculaire
MOREAU René	Mécanique
MORET Roger	Physique Nucléaire Corpusculaire
PARIAUD Jean-Charles	Chimie-Physique
PAUTHENET René	Physique du Solide - Cristallographie
PERRET René	Automatique
PERRET Robert	Electrotechnique
PIAU Jean-Michel	Mécanique
POLOUJADOFF Michel	Electrotechnique
POUPOT Christian	Electronique - Automatique
RAMEAU Jean-Jacques	Electrochimie - Corrosion
ROBERT André	Chimie appliquée et des matériaux
ROBERT François	Analyse numérique
SABONNADIÈRE Jean-Claude	Electrotechnique
MmeSAUCIER Gabrielle	Informatique fondamentale et appliquée

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

PROFESSEURS DES UNIVERSITES

Mme SCHLENKER Claire	Physique du Solide - Cristallographie
MM SCHLENKER Michel	Physique du Solide
SOHM Jean-Claude	Chimie Physique
TRAYNARD Philippe	Chimie - Physique
VEILLON Gérard	Informatique fondamentale et appliquée
ZADWORYN François	Electronique

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

M FRUCHART Robert	Directeur de Recherche
MM ANSARA Ibrahim	Maître de Recherche
CARRE René	Maître de Recherche
DAVID René	Maître de Recherche
DRIOLE Jean	Maître de Recherche
KAMARINOS Georges	Maître de Recherche
KLEITZ Michel	Maître de Recherche
LANDAU Ioan-Doré	Maître de Recherche
MERMET Jean	Maître de Recherche
MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (Décision du Conseil Scientifique)

E.N.S.E.E.G.

MM ALLIBERT Michel	
BERNARD Claude	
CAILLET Marcel	
Mme CHATILLON Catherine	
MM COULON Michel	
HAMMOU Abdelkader	
JOUD Jean-Charles	
RAVAINE Denis	
SAINFORT	
SARRAZIN Pierre	
SOUQUET Jean-Louis	
TOUZAIN Philippe	
URBAIN Georges	

C.E.N.G.

Laboratoire des Ultra-Réfractaires
ODEILLO

E.N.S.M.S.E.

MM BISCONDI Michel	
BOOS Jean-Yves	
GUILHOT Bernard	
KOBILANSKI André	
LALAUZE René	
LANCELOT Francis	
LE COZE Jean	
LESBATS Pierre	
SOUSTELLE Michel	
THEVENOT François	

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

THOMAS Gérard
TRAN MINH Canh
DRIVER Julian
RIEU Jean

E.N.S.E.R.G.

MM BOREL Joseph
CHEHIKIAN Alain
VIKTOROVITCH Pierre

E.N.S.I.E.G.

MM BORNARD Guy
DESCHIZEAUX Pierre
GLANGEAUD François
JAUSSAUD Pierre
Mme JOURDAIN Geneviève
MM LEJEUNE Gérard
PERARD Jacques

E.N.S.H.G.

M DELHAYE Jean-Marc

E.N.S.I.M.A.G.

MM COURTIN Jacques
LATOMBE Jean-Claude
LUCAS Michel
VERDILLON André

*
* *
*

Nous tenons à remercier:

Monsieur Louis BOLLIET, Professeur à l'Université Scientifique et Médicale de Grenoble, qui nous a fait l'honneur de présider le jury de cette thèse.

Monsieur Jean-François CHAMBON, responsable scientifique à l'Ecole des Mines de Saint-Etienne, auprès de qui nous avons toujours trouvé aide et encouragements.

Monsieur Jean-Claude CHUPIN, Ingénieur à CII-HB, qui a accepté de juger ce travail.

Monsieur Serge GUIBOUD-RIBAUD, Directeur du département informatique de l'Ecole des Mines de Saint-Etienne, grâce auquel ce travail a pu aboutir.

Monsieur Jean LE BIHAN, monsieur Gérard LE LANN et les membres du projet SIRIUS avec qui nous avons eu des discussions enrichissantes.

Nous exprimons également toute notre amitié à Michèle CART, Paul PAYS et Bah-Thu N'GUYEN, qui nous ont toujours apporté une aide spontanée.

Enfin nous remercions mesdames BONNEFOY et LALLICH pour l'aide précieuse et efficace qu'elles nous ont apportée pour la confection de cet ouvrage. Egalement messieurs BROSSARD, LOUBET et VELAY, qui ont assuré avec beaucoup de soins le tirage de cet ouvrage.

S O M M A I R E

<u>INTRODUCTION</u>	1
1. Les limites de l'informatique traditionnelle.....	3
2. Pour une informatique répartie.....	3
2.1. Répondre à un besoin.....	3
2.2. Des perspectives pleines de promesses.....	4
2.3. L'aide à la normalisation.....	4
2.4. But de notre étude.....	4
3. Etat actuel des systèmes répartis.....	5
3.1. Systèmes de fichiers répartis.....	5
3.2. Bases de données réparties.....	6
3.3. Deux types de solutions.....	6
4. Notre démarche.....	7
4.1. Les principes de base.....	7
4.2. La méthode employée.....	8
4.3. Aspect validation.....	8
 I - <u>CONTEXTE DE L'ETUDE : LES SYSTEMES REPARTIS</u>	 9
1. Aperçu historique.....	11
2. Environnement réseau d'ordinateurs.....	12
3. Environnement Centre de Calcul Réparti.....	15
4. Les Systèmes de Gestion de Fichiers.....	18
4.1. Aperçu général.....	19
4.2. Points communs.....	22
4.3. Incompatibilité et disparité entre SGF.....	25
5. But de l'étude : coopération d'applications.....	27

II - <u>LES SYSTEMES DE GESTION DE FICHIERS REPARTIS</u>	31
1. Nécessité d'un système de fichiers réparti.....	33
2. Système de fichiers distribués (DFS).....	34
2.1. Aperçu du réseau DCS.....	34
2.2. Structure du système DFS.....	35
3. Méthode d'accès Direct Réseau (MADRE).....	38
3.1. Environnement de MADRE.....	38
3.2. Fonctionnalité de MADRE.....	38
3.3. Mise en oeuvre de MADRE.....	40
4. Protocole d'Accès aux Données (DAP).....	40
4.1. Les différents niveaux de DAP.....	41
4.2. Description fonctionnelle de DAP.....	42
5. Le système DISCO.....	43
5.1. Généralités et but du système DISCO.....	43
5.2. Niveau de l'interface DISCO.....	44
5.3. Allocation de ressources distribuées.....	44
5.4. Gestion du catalogue des fichiers logiques..	45
III - <u>LES SYSTEMES DE GESTION DE BASES DE DONNEES REPARTIS</u>	47
1. Diverses approches de la distribution.....	49
1.1. Méthode descendante ou répartition.....	50
1.2. Méthode ascendante ou coopération.....	52
2. Analyse de quelques SGBDR.....	55
2.1. Architecture de SDD-1.....	56
2.2. Le modèle ADD.....	57
2.3. Le projet FRERES.....	59
2.4. Projet SIRIUS-DELTA.....	63

IV	- <u>MODELES D'ARCHITECTURE DE SYSTEMES</u>	67
1.	Nécessité d'un modèle commun d'architecture.....	69
2.	Le modèle ANSI/SPARC/X3.....	70
3.	Le modèle de référence de l'ISO.....	72
3.1.	Objectifs.....	72
3.2.	L'architecture à sept couches.....	74
4.	Architecture orthogonale.....	76
5.	Techniques de structuration.....	77
5.1.	Structuration par emballage.....	77
5.2.	Structuration en couches.....	79
6.	Quelques approches de la répartition de données....	82
6.1.	L'approche séparatrice.....	82
6.2.	L'approche unificatrice.....	82
6.3.	Notre approche : un palliatif.....	83
V	- <u>MECANISMES D'EXECUTION REPARTIE</u>	85
1.	Communication entre applications.....	88
2.	Langages de requêtes.....	90
2.1.	Différentes solutions au problème de traduction.....	90
2.2.	Critères de choix d'un langage de commande..	94
2.3.	Hiérarchie de niveaux.....	95
3.	Techniques réseaux.....	97
4.	Protocole de niveau session.....	98
4.1.	Services offerts.....	98
4.2.	Principe de la communication interprocessus.	99
5.	Niveau présentation.....	99
5.1.	But de la couche présentation.....	99
5.2.	Les protocoles offerts.....	99

VI - <u>LE SYSTEME DE COOPERATION DE SGF</u>	101
1. Intérêt et objectif.....	103
1.1. Augmentation potentielles des ressources....	103
1.2. Coopération de systèmes hétérogènes.....	104
1.3. Spécificité par rapport aux systèmes de fichiers distribués existants.....	104
2. Modèle de coopération choisi.....	105
2.1. Composants du SCSGF.....	106
2.2. Types de dialogues possibles.....	108
2.3. Gestion de catalogue.....	110
3. Mécanisme de fonctionnement du Système de Coopération.....	113
3.1. Localisation des fichiers.....	115
3.2. Espace de désignation.....	117
3.3. Contrôle de concurrence d'accès.....	118
4. Le Système de Coopération proposé.....	121
4.1. Caractéristique.....	121
4.2. Méthodologie.....	124
4.3. Apport du SCSGF.....	125
 VII - <u>ARCHITECTURE RESULTANTE DU SYSTEME DE COOPERATION</u> ..	127
1. Fonctionnalité du système de coopération.....	129
2. Décomposition client-serveur.....	132
3. Schéma de principe d'une architecture.....	133
4. Les entités fonctionnelles.....	134
4.1. Le processus de soumission de requêtes.....	134
4.2. L'entité client.....	136
4.3. L'entité serveur.....	136
4.4. Le gestionnaire de catalogue.....	137
5. Structure et gestion du catalogue.....	138

6. Approche fonctionnelle.....	139
6.1. Architecture distribuée du système de coopération.....	139
6.2. Environnement d'un sous-système local.....	140
7. Interactions entre les différentes entités.....	142

VIII - LE CENTRE DE CALCUL REPARTI ET SON OUTIL SCSGF....145

1. Le projet CCR.....	147
1.1. Idées directrices.....	147
1.2. Le correspondant.....	149
1.3. Les moyens de communication.....	150
1.4. Approche fonctionnelle.....	153
1.5. La fonction d'interprétation.....	155
2. Interaction entre le CCR et le SCSGF.....	156
2.1. Localisation du SCSGF dans le CCR.....	157
2.2. Exemple d'utilisation possible.....	158
3. Extension à d'autres systèmes.....	162

IX - DEFINITION DE L'EXTENSION DU CONTEXTE D'APPLICATION DE L'OUTIL SCSGF.....163

1. Motivations des réseaux d'ordinateurs.....	165
2. Contexte : les réseaux d'ordinateurs.....	166
2.1. Réseau de communication.....	166
2.2. Système réseau.....	167
3. Présentation de quelques exemples.....	168
3.1. Réseau local.....	170
3.2. Réseau général.....	172
4. Intérêt et applicabilité d'un SCSGF.....	175
5. Limitations.....	177

<u>CONCLUSION</u>	179
1. Bilan de l'étude.....	181
1.1. Principaux centres d'intérêt.....	182
1.2. Apport d'un système de coopération.....	183
2. Prospectives.....	184
2.1. Bases de données et Fichiers.....	184
2.2. Architectures des systèmes répartis.....	185
2.3. Aspect méthodologie.....	186
2.4. Conditions d'intégration de la coopération..	187

INTRODUCTION

1. Les limites de l'informatique traditionnelle
2. Pour une informatique répartie
 - 2.1. Répondre à un besoin
 - 2.2. Des perspectives pleines de promesses
 - 2.3. L'aide à la normalisation
 - 2.4. But de notre étude
3. Etat actuel des systèmes répartis
 - 3.1. Systèmes de fichiers répartis
 - 3.2. Bases de données réparties
 - 3.3. Deux types de solutions
4. Notre démarche
 - 4.1. Les principes de base
 - 4.2. La méthode employée
 - 4.3. Aspect validation

1. Les limites de l'informatique traditionnelle

L'Informatique traditionnelle a, pendant longtemps, dicté ses prérogatives et imposé ses choix à la structure des systèmes d'informations au sein des organisations. Et comme la structure du système d'informations reflète celle de l'organisation, c'est en définitive celle-ci qui a souffert des contraintes apportées par l'informatique. C'est ainsi que, très souvent, les contraintes techniques préjugent déjà du mode d'organisation et imposent celui-ci car la présence des ordinateurs accentue fréquemment les pesanteurs naturelles des entreprises et des administrations ; les procédures centralisatrices de l'informatique traditionnelle renforcent le centre aux dépens de la périphérie, et accentuent ainsi les clivages entre les échelons de direction et leurs cellules d'exécution.

2. Pour une informatique répartie

2.1. Répondre à un besoin

L'information est par nature répartie dans les organisations (LE BIHAN: BDR-12). C'est pour cela qu'elle s'accommode mal de la structure très centralisatrice de l'informatique traditionnelle dans laquelle les impératifs

de rentabilité économique ont conduit à rechercher une économie d'échelle par une concentration sans rapport avec les besoins propres aux organisations pour aboutir à des coûts moindres des moyens de mémorisation et de traitement de l'information (ZIMMERMANN: RES-40).

2.2. Des perspectives pleines de promesses

Par contre l'informatique répartie semble bénéficier d'un avenir plus prometteur grâce à la micro-électronique (microprocesseurs) et aux progrès faits en matière de réseaux d'ordinateurs, car elle est plus conforme à la répartition naturelle de l'information. Le terme téléinformatique déjà devenu classique fait de plus en plus place à celui de télématique.

2.3. L'aide à la normalisation

Dans ce contexte de systèmes répartis, il devient de plus en plus nécessaire d'élucider les problèmes posés par l'organisation des réseaux informatiques pour arriver à des standards et des normes afin de réduire la dépendance de l'utilisateur vis-à-vis d'un constructeur particulier. En effet, ce qu'on attend du service informatique c'est, non pas qu'il s'érige en "micro-pouvoir" à l'intérieur d'une organisation" (LORRAINS: RES-23), mais qu'il reste un outil aussi commode d'emploi que possible. Ainsi les mécanismes mis en oeuvre, aussi complexes soient-ils, doivent rester transparents au moment de l'utilisation.

2.4. But de notre étude

C'est dans le cadre des systèmes répartis que se

place notre étude qui veut apporter une contribution à l'architecture de ces systèmes. Trouvant sa motivation dans un problème concret, elle procède de l'analyse de la classe de solutions et d'une problématique plus vaste à laquelle sont confrontés la plupart des concepteurs de systèmes répartis. Cette problématique pourrait se résumer ainsi :

Comment faire profiter l'utilisateur final d'une certaine répartition spatiale possible des ressources d'un système informatique en le dispensant de l'embarras que constitue pour lui la gestion très complexe de cette répartition ?

En effet, un des problèmes auxquels nous avons été confrontés dans le cadre du projet du Centre de Calcul Réparti (fondé sur un réseau de mini-ordinateurs hétérogènes) de l'Ecole des Mines (chapitres I et VIII) a été de mettre en place des mécanismes qui puissent offrir à chaque utilisateur un environnement personnalisé de ressources et de services en lui épargnant le souci de connaître la localisation exacte de ces ressources et de ces services. Dès lors pour assurer cette transparence, il s'avérait nécessaire, entre autres, d'assurer la transparence de la localisation des ressources FICHIERS. Le fichier est en effet l'un des rares moyens d'échange de données et d'informations entre services qui puisse être soit permanent soit provisoire selon l'usage qu'on veut en faire.

3. Etat actuel des systèmes répartis

3.1. Systèmes de fichiers répartis

Une analyse rapide (chapitre I) des différents Systèmes de Gestion de Fichiers (SGF) standards existant à

l'heure actuelle permet de mettre en évidence leur manque d'ouverture sur l'extérieur et leur inadéquation presque totale à une quelconque prise en compte d'un environnement réparti.

3.2. Les Bases de Données Réparties

Le même genre de problème se pose également dans les Systèmes de Gestion de Bases de Données proposés par les constructeurs dès lors qu'on aborde la répartition de l'espace de stockage. Ces problèmes sont d'autant plus épineux qu'on veut répartir les fonctions de manipulation (décomposition de requêtes, optimisation des chemins d'accès, exécution répartie...). Aussi bien dans les systèmes de fichiers répartis (chapitre II) que dans les bases de données réparties (chapitre III), la nécessité de communication entre entités distantes ajoute un problème de délais de propagation sans oublier les problèmes de désignation.

3.3. Deux types de solutions

Pour remédier à toutes ces lacunes deux catégories de solutions ont été explorées.

La première consiste à faire table rase de tout le logiciel constructeur et à concevoir de toutes pièces un système ouvert qui puisse s'intégrer facilement dans un environnement réparti.

La deuxième démarche est plus pragmatique en ce sens qu'elle garde les logiciels des constructeurs et qu'elle essaie de les intégrer à un environnement réparti.

Pour arriver à ce but, il faut ajouter une interface qui étend les possibilités des systèmes initiaux et leur offre une ouverture sur l'extérieur.

Bien sûr, aucune de ces deux catégories de solutions n'est une panacée. Mais, il semble intéressant d'analyser les avantages des unes et des autres pour en tirer profit.

4. Notre démarche

Nous avons cherché à définir, dans un contexte réseau, les mécanismes qui permettent de gérer, de façon transparente à l'utilisateur, la répartition de l'espace de stockage global (afin de l'optimiser en particulier). Une des principales contraintes que nous nous sommes imposées a été de modifier aussi peu que possible le logiciel constructeur afin d'en garantir la pérennité.

4.1. Les principes de base

En nous fondant sur le modèle d'architecture défini par l'ISO (ISO: MAS-12), nous avons défini des mécanismes généraux de coopération d'applications en environnement hétérogène (chapitre IV) sans faire trop d'hypothèses restrictives sur la nature du réseau et les couches inférieures à la couche session (transport, réseau, liaison de données, physique). L'objet de cette partie de notre étude est d'arriver à un modèle fonctionnel, aussi général que possible.

4.2. La méthode employée

Aussi avons-nous été conduits à définir (chapitre VI) une interface entre les applications des utilisateurs et les différents systèmes de gestion de fichiers dans le but de faire coopérer les différents SGF des sites participants.

Mais pour ce faire nous avons dû élargir le problème de la transparence de la localisation des ressources à d'autres types d'applications comme les systèmes de données distribuées (fichiers répartis, méthodes d'accès réseaux en général) ou les Systèmes de Gestion de Bases de Données Réparties (SGBDR) qui demeurent encore un domaine de recherche, cependant très riche en enseignements dans l'étude et la formalisation des problèmes généraux des systèmes répartis : contrôle de concurrence d'accès, données dupliquées, allocation optimale de ressources en environnement réparti, mécanismes d'exécution répartie (chapitre V).

4.3. Aspect validation

L'application (chapitre VIII) à une architecture existante, en l'occurrence celle du Centre de Calcul Réparti, doit offrir un terrain de validation des idées précédemment émises. L'étude de l'applicabilité à d'autres systèmes doit servir à compléter cet aspect validation et utilité de l'étude entreprise.

L'applicabilité à d'autres systèmes (chapitre IX) nous donne l'occasion d'analyser les différentes autres utilisations possibles du modèle de coopération sur différents types de réseaux d'ordinateurs. La prospective nous offre aussi l'opportunité d'avancer des hypothèses sur les axes de développements futurs des systèmes répartis en général et des coopérations d'applications (comme les fichiers répartis ou les bases de données réparties) en particulier.

CHAPITRE I

CONTEXTE DE L'ETUDE : LES SYSTEMES REPARTIS

1. Aperçu historique
2. Environnement réseau d'ordinateurs
3. Environnement Centre de Calcul Réparti
4. Les Systèmes de Gestion de Fichiers
 - 4.1. Aperçu général
 - 4.2. Points communs
 - 4.3. Incompatibilité et disparité entre SGF
5. But de l'étude : coopération d'applications

1. Aperçu historique

L'outil informatique était, à ses débuts, essentiellement orienté vers les applications scientifiques complexes. Le volume des données à traiter était en effet très faible comparé à la complexité du traitement à effectuer sur ces données.

Puis en suivant deux pôles d'attraction bien distincts (MIRANDA: RES-28):

- d'une part l'accroissement du volume des données à traiter,
- d'autre part la tendance à la décentralisation de certaines fonctions (acquisition de données par exemple),

l'évolution a tout naturellement conduit aux réseaux d'ordinateurs.

En effet l'accroissement considérable du volume de données à traiter a nécessité de plus en plus une intelligence de traitement beaucoup plus importante et une mise en commun de ressources distribuées.

Répondant aux vœux d'un nombre d'utilisateurs de plus en plus important, l'informatique, en même temps qu'elle se diversifiait s'introduisait dans des milieux très divers.

Dès lors commence l'ère de la téléinformatique, résultat de la rencontre de l'informatique et des télécom-

munications. Ainsi nous assistons au développement des réseaux d'ordinateurs et même des réseaux publics de transmission de données.

Après une première période caractérisée surtout par des progrès rapides dans la maîtrise des techniques réseaux, l'évolution a été marquée par le désir croissant d'avoir des applications qui puissent s'exécuter sur l'environnement ainsi constitué.

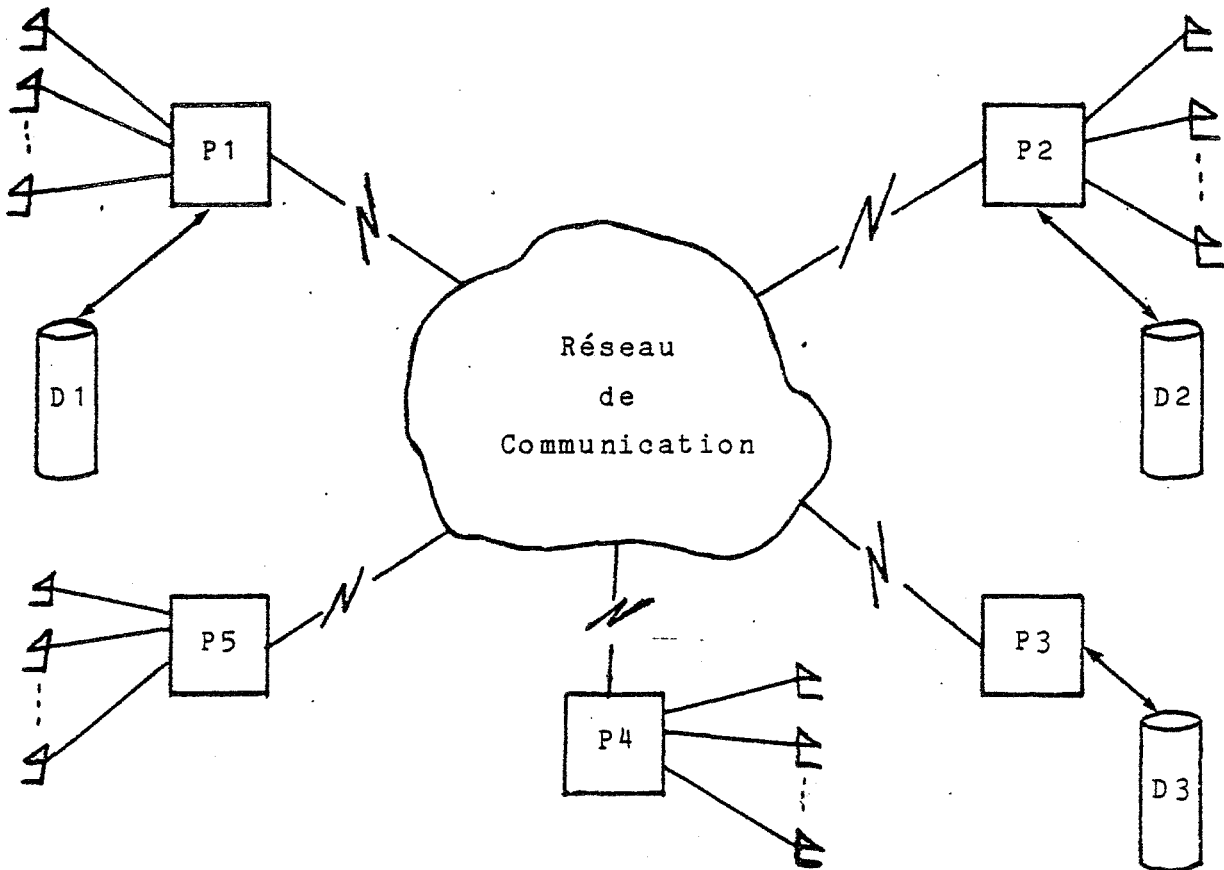
2. Environnement réseaux d'ordinateurs

Une entreprise ayant fait l'option d'acquérir progressivement, selon ses besoins, des moyens informatiques capables de résoudre ses problèmes là où ils se posent, aura certainement un système informatique efficace. Néanmoins, il sera dispersé et isolé dans les différents centres d'activité de l'entreprise, ce qui le rendra redondant, donc coûteux, et probablement peu performant.

Dans un système dispersé, les moyens informatiques sont forcément rassemblés en groupes capables de constituer des points d'échange (figure - 1), uniquement, sans qu'ils soient interconnectés entre eux. Cela veut dire, qu'un tel Centre de Calcul, si petit soit-il, sera complètement autonome; d'où sa redondance, lorsqu'on multiplie ce système au sein de l'entreprise.

L'utilisation optimale d'un tel système passe par l'interconnexion de tous ses moyens informatiques au niveau fonctionnel. Le réseau ainsi créé permettra à partir d'un centre d'activité, l'accès à n'importe quels autres centres d'activités de l'entreprise (LE BIHAN: BDR-12), (LE LANN, NEGARET: MER-10). Nous passons donc d'un système éparpillé à

un système réparti en ce sens qu'au niveau fonctionnel, avec un certain recul, le système est vu comme s'il n'avait qu'un seul service, la "gestion de l'entreprise", réparti sur les sites qui le composent.



P1 et P2 sont des points d'échange: présence de données et utilisateurs.

P3 est un point de stockage: présence de données uniquement.

P4 et P5 sont des points d'accès: présence d'utilisateurs uniquement.

Figure - 1.

La topologie d'un tel système doit, en général, éliminer un certain nombre de redondances. Il peut être composé de points d'accès, de stockage et d'échange (figure - 1), situés dans les différents centres d'activités de

l'entreprise (SPACCAPIETRA: BDR-16). Ces systèmes répartis fondés sur un réseau local peuvent avoir comme caractéristique, un environnement homogène ou hétérogène:

a) Réseau homogène: c'est la plupart du temps le système fourni par un constructeur unique. Etant donné la compatibilité qui existe entre les ordinateurs, malgré l'existence éventuelle de différents modèles, la communication et le transfert de données entre eux se fait de façon quasi transparente. En effet, même les données type "programme objet" peuvent être transférées et exécutées, d'une machine à l'autre, pratiquement sans restrictions.

b) Réseau hétérogène: c'est le système composé de machines fournis par différents constructeurs. Dans un tel réseau, la communication et le transfert de données entre sites constituaient auparavant un problème considérable. Aujourd'hui, les efforts de normalisation d'un certain nombre de techniques, qui ont déjà fait leur preuve, l'ont rendu partiellement soluble.

Le succès des réseaux hétérogènes, s'explique par plusieurs raisons dont nous mentionnons quelques unes des plus importantes:

- assez grande indépendance vis-à-vis de chaque constructeur;
- possibilité d'avoir le(s) mini ordinateur(s) qu'il faut pour résoudre le(s) problème(s) spécifique(s) de l'entreprise;
- etc.

Une autre raison, et peut-être la plus courante, est historique: c'est celle de nombreuses entreprises qui se sont regroupées pour une raison ou pour une autre alors qu'elles avaient déjà leurs systèmes informatiques. Celles-ci éprouvent donc le besoin de faire communiquer, ou coopérer, ces différents systèmes afin de faciliter la gestion et optimiser l'utilisation des différentes ressources du réseau ainsi créé.

Néanmoins, nous constatons que les avantages attendus du partage des ressources, ressource fichiers par exemple, dépendent beaucoup de la coopération existant entre les services responsables de leur gestion. Dans le cas des fichiers ces avantages dépendent du Système de Gestion de Fichiers existant sur chaque site, fourni par chacun des constructeurs.

3. Environnement Centre de Calcul Réparti

Le développement de réseaux généraux d'ordinateurs (dont le précurseur fut le réseau ARPA) a ouvert l'ère de l'informatique répartie. Dans cette optique, le Projet Pilote de Réseau Général CYCLADES (Chapitre IX §3.2), a été lancé en 1972. Pour offrir aux matériels du Centre de Calcul de l'Ecole de Mines (ordinateur P1175 et terminaux) un accès au réseau CYCLADES, un logiciel de frontal a été développé sur un mini-ordinateur (T1600 de Télémécanique) de l'Ecole (CHAMBON, LE BIHAN: RES-6).

Elargissant le contexte initial, on envisagea de généraliser le rôle du Frontal et d'y concentrer les fonctions de banalisation des terminaux et le logiciel de communication avec le réseau externe. Il permettrait ainsi :

- aux terminaux locaux d'accéder aux services distants disponibles sur le réseau CYCLADES;
- aux terminaux distants (ceux du réseau CYCLADES) d'accéder aux services locaux.

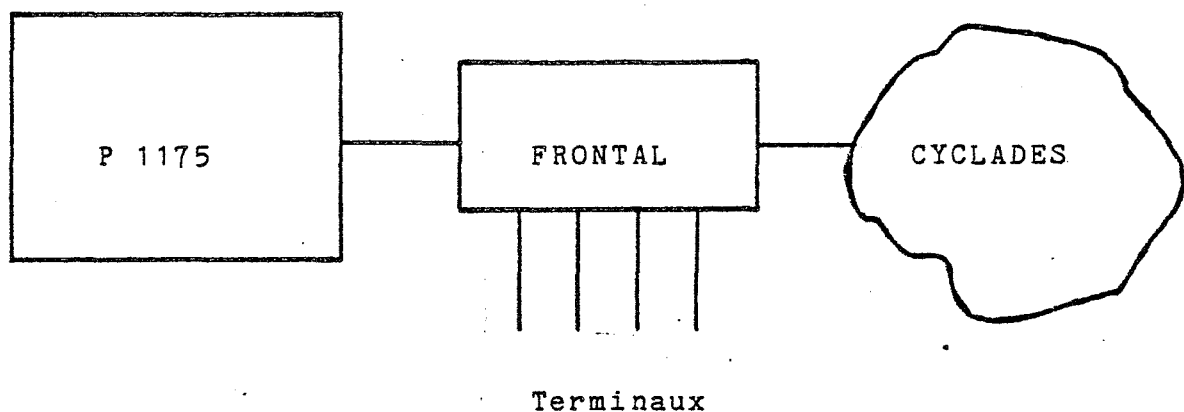


Figure - 2. Frontal en environnement téléinformatique.

Ainsi est née l'idée du Centre de Calcul Réparti (CCR), fondé sur un réseau de mini-ordinateurs hétérogènes. Il s'agit de fournir à l'utilisateur final des ressources, matériel et logiciel, des services et non un ensemble d'ordinateurs (CHAMBON et al.: RES-7). En effet il ne faudrait pas que la mise en place du réseau de mini-ordinateurs ait pour corollaire le fait que l'utilisateur se retrouve confronté à une multitude de systèmes d'exploitation différents dont il ne peut raisonnablement pas espérer avoir la maîtrise (CART: RES-5).

La stratégie adoptée pour la mise en place du CCR profite du développement des techniques réseaux qui met à la portée des utilisateurs des outils plus conformes à la

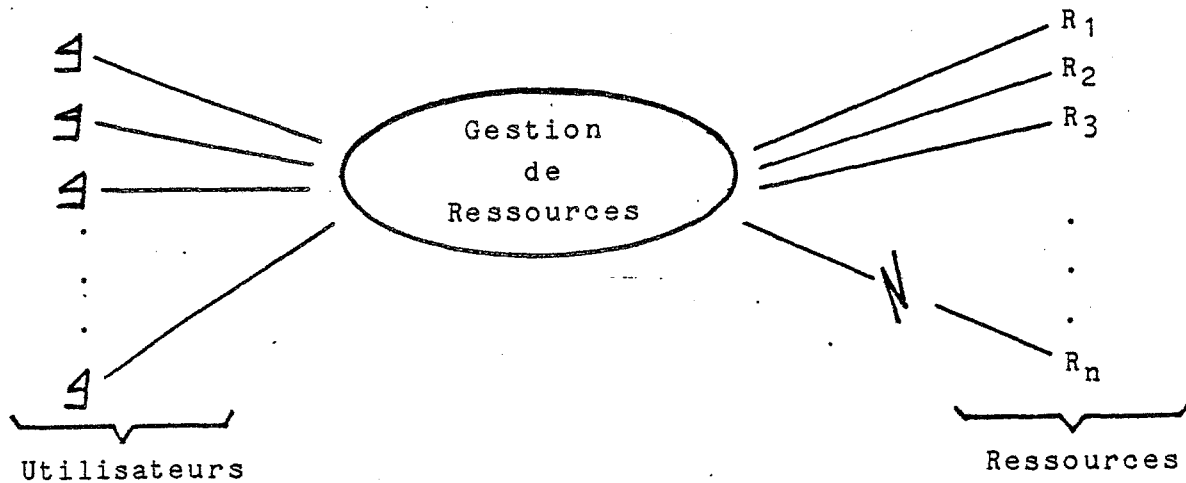


Figure - 3. Structure logique du CCR.

nature répartie des sources d'information et à la multiplicité des centres de décision. En outre la mise en commun des ressources informatiques répond mieux à l'objectif de partage des informations.

La structure logique apparente à l'utilisateur du CCR est composée:

- de ressources bien définies ayant chacune une fonction bien précise;
- d'une entité qui gère ces ressources et à qui l'utilisateur s'adresse d'une façon unique indépendamment de la localisation de ces ressources;
- d'un moyen pour l'utilisateur de communiquer avec le Gestionnaire des Ressources: souvent par l'intermédiaire d'un terminal.

L'aspect physique intéresse peu l'utilisateur, de même que les détails de réalisation. Peu lui importe si cette structure logique est virtuelle et simulée pour lui cacher les vrais mécanismes. L'important pour l'utilisateur, c'est de pouvoir accéder aux ressources offertes de façon unique et indépendamment de leur localisation (JUREMA, NIANG: RES-22). La conséquence en est qu'il faut lui donner l'apparence d'un seul calculateur qui matérialiserait pour lui le Centre de Calcul Réparti avec (CART: RES-5) :

- le catalogue des services offerts par le Centre,
- les ressources périphériques du Centre,
- un langage de commande unique et simple .

Dans cette optique, la prise en compte des ressources fichiers est un point important. En effet, l'application fichier est un service particulier, fourni par le système d'exploitation de chaque site. L'image qu'on veut donner à l'utilisateur dépend donc de la coopération entre ces différents services : les Systèmes de Gestion de Fichiers (SGF).

4. Les Systèmes de Gestion de Fichiers

Un fichier est une collection d'articles, c'est à dire d'éléments d'information de tailles fixes ou variables, et identifiée par un nom. Il est conservé sur un support matériel dont les principaux types sont:

- les cartes et rubans perforés,
- les bandes, tambours et disques magnétiques,
- etc.

L'avance technologique et l'augmentation du volume d'information à traiter ont imposé des changements quant au support physique utilisé et à la façon de stocker l'information. Après une période d'utilisation intense, les cartes perforées, dont le volume physique des fichiers est généralement peu important et les mises à jours assez lentes, ont été remplacées par des bandes et disques magnétiques. L'avantage de ces supports, notamment en temps d'accès et volume physique des fichiers, a fait d'eux le moyen de stockage le plus couramment utilisé actuellement. Pour des raisons de simplicité, dans les paragraphes suivants le support physique considéré sera, toujours, le disque magnétique, l'extension à d'autres supports ne constituant pas un problème majeur.

4.1. Aperçu général

Les SGF (Systèmes de Gestion de Fichiers) se proposent de fournir à l'utilisateur, des outils d'accès à l'information. Leur fonction de base est de répondre aux demandes d'information stockées en mémoire secondaire, et de transférer l'information entre une mémoire secondaire et l'utilisateur, ou son application. L'utilisateur traite l'information et le SGF gère son contenant. Ce qui représente pour l'utilisateur un paquet d'information et a une certaine identification, représente pour le SGF une boîte et garde la même identification. Eventuellement une identification pourra être associée à un catalogue, fichier, article, etc.

Il est souhaitable de rendre les programmes d'application indépendants des caractéristiques techniques d'une mémoire secondaire. Ainsi les SGF offrent à leurs utilisateurs, quel que soit le support physique utilisé, la possibilité d'adresser, en lecture ou écriture:

- l'unité d'information (généralement le mot) ou
- quelques milliers d'unités d'information.

A la création d'un fichier, le SGF alloue à son créateur le nombre d'unités d'information sollicité. Selon

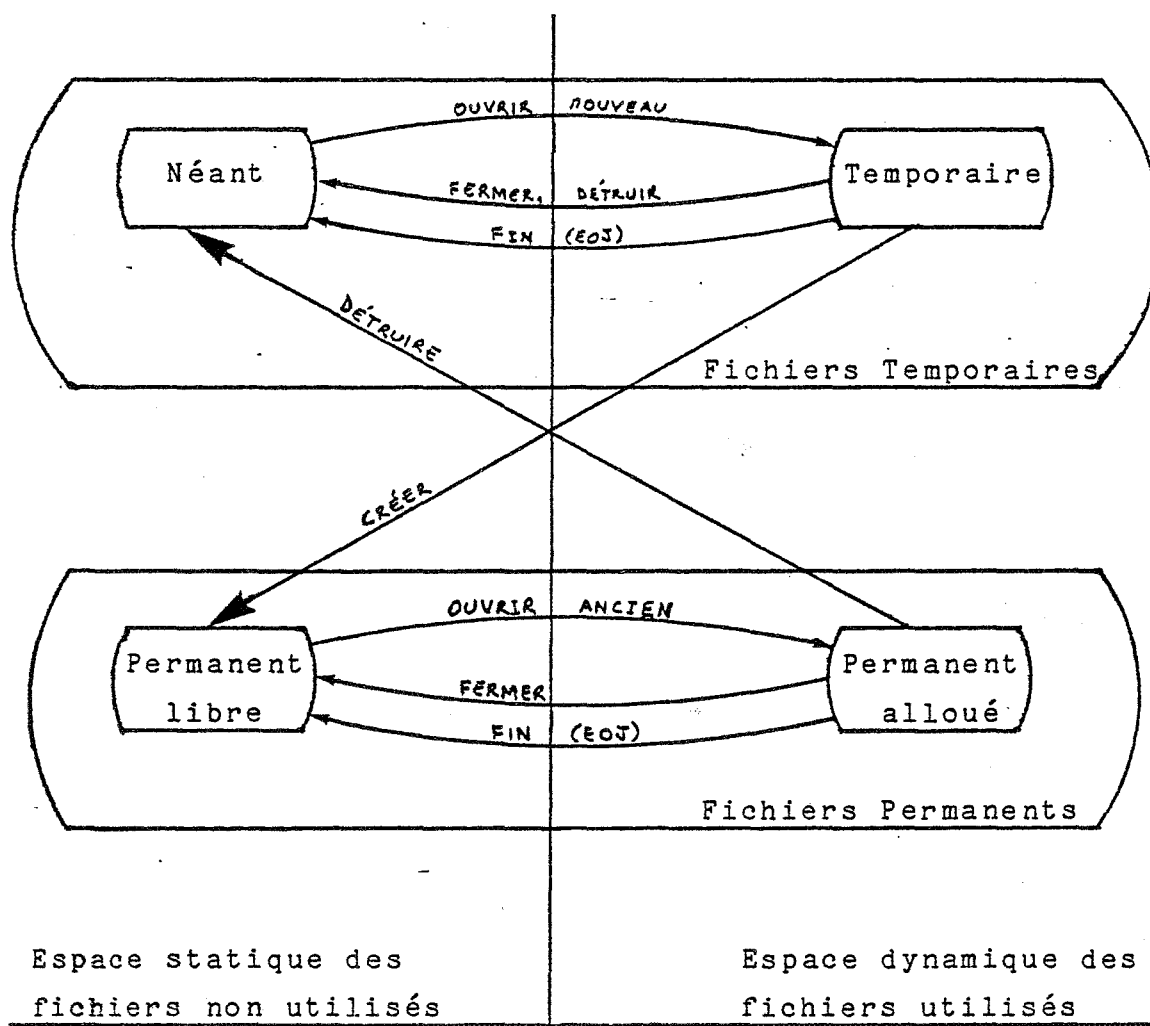


Figure - 4. Vision schématique de la vie d'un fichier.

l'utilisation qui sera faite de ce fichier, celles-ci pourront être libérées peu de temps après (fichier temporaire) ou être conservées plus longtemps (fichier permanent). Au moment de la création d'un fichier, l'utilisateur doit préciser la façon dont il veut que son fichier soit organisé et accédé. Les SGF proposent des méthodes d'organisation des informations (ensemble des règles régissant la structure interne d'un fichier) et des méthodes d'accès à ces informations (techniques de traitement de fichiers pour transférer les données entre le support physique de stockage et la mémoire centrale), adaptées à chaque type de fichier.

L'utilisateur peut créer un nouveau fichier, lire, mettre à jour ou en détruire un qui existait déjà, pendant sa session de travail. A certaines de ces actions correspondent des demandes ou libérations de place en mémoire secondaire. Le SGF doit résoudre ces problèmes d'allocation et de libération de place afin d'éviter que la mémoire secondaire soit trop vite saturée. Autrement dit, il doit prendre à sa charge la gestion de l'espace de stockage mis à la disposition de l'utilisateur (les contingences matérielles et économiques actuelles empêchent de conserver toute l'information en mémoire centrale).

Les informations concernant les caractéristiques de chaque fichier (nom, emplacement en mémoire secondaire, taille, protection, etc.) et la gestion de l'espace de stockage (nombre et localisation des places libres ou occupées) sont souvent maintenues dans une structure hiérarchisée en arbre (MAHL, BOUSSARD: GEN-6), (BERZTISS: GEN-1), (GOTLIEB, GOTLIEB: GEN-5). Les SGF placent ces informations dans des catalogues, tables ou listes, utilisées pour la gestion des fichiers qu'ils abritent. Le gestionnaire de ces tables constitue un des principaux composants des SGF.

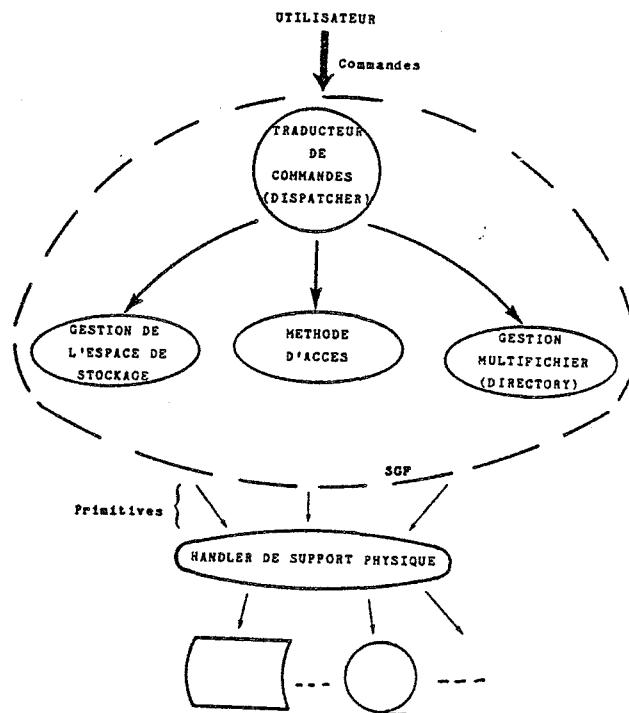


Figure - 5. Schéma fonctionnel d'un SGF.

4.2. Points communs

Lorsqu'un utilisateur manifeste au SGF son intention de travailler sur un fichier donné, le SGF après confrontation entre les droits de l'utilisateur et les caractéristiques du fichier, lui crée une table ou catalogue. Ceci a pour but d'assurer la protection du fichier contre des actions imprévues susceptibles de se produire pendant la session de travail. Cette protection est assurée vis-à-vis des trois principaux problèmes suivants, à savoir (GARDARIN, GOMEZ: SGF-5):

1) accès concurrents: la plupart des Systèmes de Gestion de Fichiers offrent la possibilité à plusieurs utilisateurs d'accéder simultanément à un même fichier au moins en lecture. Ces accès nécessitent une certaine synchronisation en cas de mises à jour sur le fichier.

2) résistance aux pannes: un fichier a une durée de vie supérieure à la durée d'exécution du programme qui l'a créé. De ce fait, le SGF doit assurer au mieux la sauvegarde des fichiers et catalogues lors des pannes logicielles ou matérielles;

3) confidentialité: elle s'applique à la protection des fichiers contre les accès non autorisés ou mal intentionnés. Pour cela le créateur d'un fichier dispose en général de la possibilité d'étendre ou restreindre son utilisation à d'autres utilisateurs ou groupes d'utilisateurs, en leur attribuant en plus, des droits d'accès différents.

La structure interne du fichier est régie par un ensemble de règles constituant l'organisation du fichier. La façon dont les articles du fichier sont accédés est la méthode d'accès.

On distingue deux méthodes d'accès principales, la méthode séquentielle, où l'utilisateur doit, pour accéder à un article donné, accéder à tous les articles précédant celui-ci dans le fichier, et la méthode indexée, où il est possible d'accéder directement à un article à partir d'une clé d'accès.

Les différentes possibilités d'organiser un fichier ont pour but de minimiser le temps d'accès à l'information. D'une manière générale elles varient peu d'un ordinateur à un autre (DEC: GEN-3, SGF-2), (CII-HB: SGF-1), (SEMS: SGF-8). Les principales manières d'organiser un fichier, sont:

a) séquentielle: les articles sont enregistrés les uns après les autres dans l'ordre d'arrivée, à partir du

début du fichier. Il existe plusieurs variantes de cette organisation:

- séquentielle partitionnée: permet de découper logiquement un fichier séquentiel en plusieurs sous-ensembles indépendants;
- séquentielle chaînée: elle utilise une zone de débordement qui permet d'ajouter des articles en milieu de fichier. Chaque article comprend un mot de chaînage;
- etc.

b) séquentielle indexée: au fur et à mesure que les articles sont enregistrés dans le fichier, l'identification de chacun ainsi que son adresse relative sont écrites dans une table, appelée index;

- séquentielle indexée triée: le fichier est divisé en parties, chaque partie étant composée de pages principales et de pages secondaires réservées pour les débordements des pages principales. A l'intérieur des pages les articles sont triés par ordre croissant des clés. Un index des pages est construit pour chaque partie et un index des parties est également construit pour chaque fichier. Tous les index sont triés par clés croissantes;

- séquentielle indexée régulière: les débordements sont traités de telle sorte que le taux de remplissage de chaque page soit le plus près possible du taux moyen. Le traitement des débordements constitue la seule différence entre cette organisation et la précédente;

c) aléatoire: une fonction particulière ("hachage" par exemple), utilisant la clé de chaque article, donne le numéro de la page dans laquelle est enregistré l'article. A l'intérieur de celle-ci, les articles sont rangés séquentiellement et une zone de débordement est prévue pour les articles excédentaires.

L'aspect organisation du fichier et méthode d'accès constitue l'un des points importants sur lesquels les constructeurs, vis-à-vis des utilisateurs, ont adopté des approches très voisines. Certains utilisent parfois une terminologie différente mais en restant fidèle au principe commun. Toutefois, il n'est pas moins vrai que cette méthodologie commune reste bien limitée au niveau externe, c'est à dire aux concepts présentés à l'utilisateur. En effet, au niveau interne, les réalisations peuvent être très différentes d'un constructeur à un autre.

4.3. Incompatibilité et disparité entre SGF

Les différences existent moins au niveau externe, comme nous l'avons montré (en raison de la standardisation des langages de programmation : COBOL, FORTRAN, PASCAL ...), qu'au niveau interne, c'est à dire dans la façon dont les SGF agissent lorsqu'ils sont sollicités par l'utilisateur. Cette différence est plus ou moins accentuée d'un constructeur à un autre selon la technique employée pour réaliser la vision conceptuelle commune offerte aux utilisateurs. En effet si les constructeurs se sont mis à peu près d'accord quant à la façon d'organiser et d'accéder un fichier, malgré l'absence de normes strictes dans ce domaine, ce n'est pas du tout le cas pour la gestion des informations nécessaires à cela.

Comme les fichiers ont des tailles différentes, les espaces qui leur sont alloués ne sont généralement pas contigus sur une mémoire secondaire. Le SGF doit pouvoir retrouver les régions composant un fichier. Pour cela il peut, soit garder la liste des régions allouées à un fichier dans une table, soit les chaîner, c'est à dire mettre dans chaque région l'adresse de la région suivante.

A chaque sollicitation de l'utilisateur, soit en lecture ou en écriture, le SGF a souvent à manipuler des tables, descripteurs ou pointeurs. Prenons par exemple le cas d'une organisation séquentielle chaînée, sur disque magnétique. Plusieurs articles de différents fichiers peuvent être enregistrés sur une même piste. Toutefois, la seule relation qui existe entre ces enregistrements est qu'ils sont consécutifs sur le support (figure 6) (JOUFFROY, LETANG: SGF-6). Il s'avère que la façon dont ont été chaînés les articles du même fichier, la localisation des pointeurs par exemple, est absolument transparente à l'utilisateur.

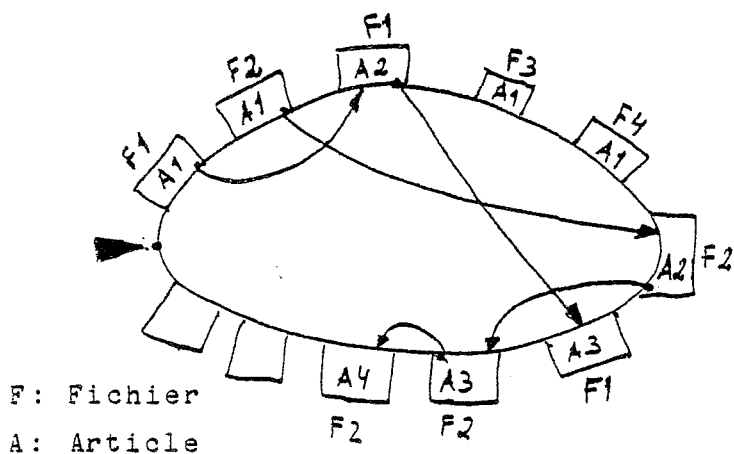


Figure - 6.

Etant donné qu'il n'était pas prévu de fournir aux utilisateurs l'accès à ces informations, tables, descripteurs ou pointeurs, leur traitement est fait de façon très hermétique sans considérations autre que l'efficacité. Néanmoins avec le développement de la téléinformatique et des minis et micro ordinateurs, l'informatique a de plus en plus d'utilisateurs désireux de mieux connaître les outils afin de pouvoir les particulariser à leurs besoins. Or, si cela n'a pas été prévu ou voulu par les constructeurs, comme c'est le cas actuellement, l'obstacle à surmonter est considérable.

Un exemple classique et qui touche un très grand nombre d'utilisateurs est celui du "formatage" des mémoires secondaires. Ceci empêche même une simple lecture du matériel d'un constructeur par un autre. Avant tout usage, le support physique doit obligatoirement être "formaté" c'est à dire, une correspondance entre quelques octets ou bits bien précis et les emplacements sur le support physique est établie. Le SGF connaîtra ainsi l'identification du volume, sa division logique, l'état (libre ou occupé) des emplacements qu'il gère, la localisation du Catalogue des fichiers, etc.

Il suffit que ces informations, généralement situées en tête du volume, soient décalées d'un bit, aient des champs intervertis ou de plus ou de moins, etc., pour que le support soit incompréhensible à un système différent de celui sur lequel il a été formaté.

Il existe actuellement des utilisateurs qui disposent de plusieurs ordinateurs, interconnectés entre eux, et qui auraient bien aimé que ces différents SGF se connaissent pour pouvoir ainsi les voir coopérer. Il se trouve que les disparités et incompatibilités existant sur certains points, entre les SGF, demandent un effort non négligeable pour aboutir à un niveau de coopération souvent insignifiant. C'est une adaptation de ce genre, c'est à dire une coopération, qui constitue l'objectif principal de cette étude; cet objectif est présenté dans le paragraphe suivant.

5. But de l'étude : coopération d'applications

Notre objectif principal, dans cette étude, est de présenter une architecture, c'est à dire un modèle, tel que les différents sous-systèmes l'intégrant puissent coopérer

entre eux en s'appuyant sur les Systèmes de Gestion de Fichiers locaux (JUREMA, NIANG: RES-22).

Nous préférons parler de coopération d'applications, car dans le cadre du CCR, il s'agit d'applications existantes pouvant fonctionner de façon autonome. Chacune de ces applications coopérantes est associée à un site d'exécution formant ainsi un composant de l'application globale.

Le "service-réseau fichier" est rendu par une application qui est répartie. Chaque partie du service s'appuie sur la gestion de fichiers standards fournie par le constructeur. De l'hétérogénéité des machines qui composent le réseau, il résulte une hétérogénéité des Systèmes de Gestion de Fichiers (SGF) et des systèmes d'exploitation.

La banalisation des fichiers constitue un avantage important pour l'utilisateur. De façon transparente il pourra accéder, à partir de son terminal, à des fichiers situés sur n'importe quel site participant au système. Ceci doit être obtenu grâce à la coopération entre les différents SGF. Il ne s'agit pas, pour nous, de construire un SGF réparti sur tous les sites et encore moins de construire un Système de Gestion de Bases de Données (FORESTIER: GEN-4), (SPACCAPIETRA: BDR-16).

A l'origine, notre étude a été conçue pour des Réseaux Locaux concernés par une mise en commun des ressources fichiers des ordinateurs qui les composent (CHAMBON et al.: RES-8, RES-9). Ces ressources sont directement accessibles de l'extérieur, chacune par son nom, sans problème d'homonymie locale au site ou au réseau.

Nos objectifs sont l'optimisation de la gestion d'espace disque et la banalisation des fichiers gérés par les différents SGF, tout en modifiant aussi peu que possible le logiciel du constructeur pour en conserver la compatibilité avec les versions ultérieures. De plus, nous devons évidemment offrir un mécanisme de sécurité et de protection des données contenues dans les fichiers.

La mise en place d'un Système de Coopération de Gestions de Fichiers est un premier pas en même temps qu'un moyen pour parvenir à une coopération plus vaste et plus générale. Cette coopération passe nécessairement par le développement d'un ensemble hiérarchisé de protocoles et d'outils rigoureux seuls gages de réussite dans les architectures des systèmes répartis.

CHAPITRE II

LES SYSTEMES DE GESTION DE FICHIERS REPARTIS

1. Nécessité d'un système de fichiers réparti
2. Système de fichiers distribués (DFS)
 - 2.1. Aperçu du réseau DCS
 - 2.2. Structure du système DFS
3. Méthode d'accès Direct Réseau (MADRE)
 - 3.1. Environnement de MADRE
 - 3.2. Fonctionnalité de MADRE
 - 3.3. Mise en oeuvre
4. Protocole d'Accès aux Données (DAP)
 - 4.1. Les différents niveaux de DAP
 - 4.2. Description fonctionnelle
5. Le système DISCO
 - 5.1. Généralités et but du système
 - 5.2. Niveau de l'interface
 - 5.3. Allocation de ressources distribuées
 - 5.4. Gestion du catalogue des fichiers logiques

1. Nécessité d'un système de fichiers réparti

L'étude des différents Systèmes de Gestion de Fichiers standards (chapitre I) permet de constater leur grande disparité et l'incompatibilité qui en résulte.

L'absence de compatibilité, ajouté au fait que les différents SGF sont conçus pour travailler presque exclusivement en environnement local font qu'ils se révèlent mal adaptés à l'utilisation en environnement réparti.

Face à ce problème, différentes solutions ont été apportées, solutions qu'il convient d'analyser pour en extraire l'essence.

Comme nous l'avons déjà mentionné en introduction, les concepteurs de systèmes ont suivi deux principaux axes.

D'aucuns, considérant qu'il est plus commode de concevoir tout un ensemble de logiciels intégrés pouvant répondre aux nécessités des systèmes répartis, ont préféré partir soit de machines nues ou de machines n'ayant qu'un noyau de système d'exploitation.

D'autres, ne voulant pas substituer un nouveau logiciel à celui du constructeur, préfèrent lui ajouter un habillage présentant vers l'extérieur une interface qui étende ses possibilités et l'intègre à l'environnement réparti.

Nous allons étudier des systèmes de fichiers distribués offrant :

- soit l'accès à distance à un fichier,
- soit l'accès à des ressources distantes,
- soit des protocoles de transferts de fichiers.
- soit enfin la gestion d'une base de fichiers considérée comme une ressource globale à plusieurs points d'entrée.

Cependant nous regrouperons ces différents systèmes suivant un découpage qui obéit à la démarche adoptée.

2. Systèmes de fichiers distribués (DFS) (FARBER: GFR-4)

2.1. Aperçu du réseau DCS

Le réseau expérimental DCS (Distributed Computer System) a été un des précurseurs des réseaux locaux et il a par conséquent beaucoup inspiré les concepteurs de ces derniers. Il est constitué de noeuds reliés entre eux et formant un anneau. Cet anneau est le système de communication.

Pour accroître la fiabilité du système, le contrôle est réparti. En outre, les protocoles de communication et d'allocation de ressources sont conçus de façon à réduire au minimum les fonctions de contrôle.

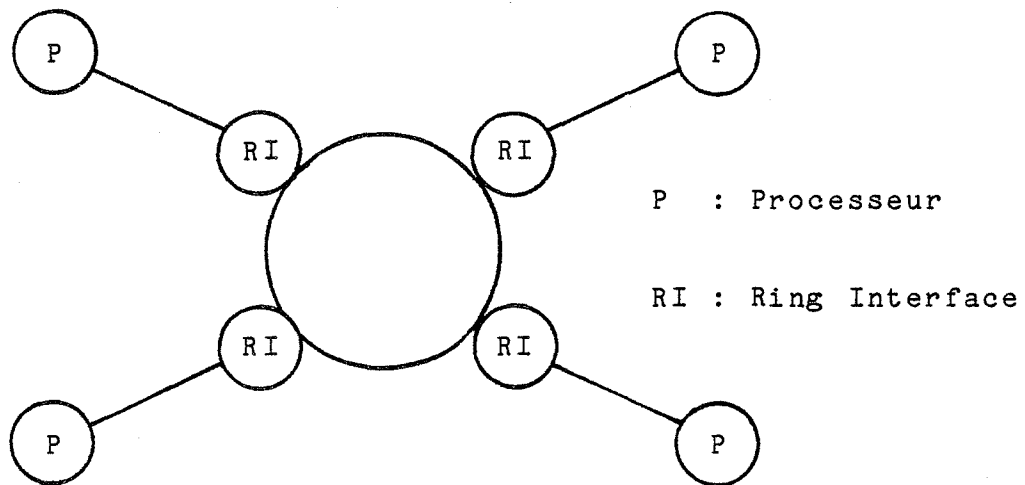


Figure - 1. Architecture en anneau ("Ring Network").

2.2. Structure du système DFS

L'architecture du système DFS ("Distributed File System") repose sur le réseau DCS dont la particularité est que les messages au lieu d'être adressés à des processeurs physiques sont au contraire adressés à des processus. La conséquence essentielle en est que l'émetteur d'un message n'a pas besoin de connaître la localisation physique du processus destinataire.

Le système de fichiers est conçu comme une application de ce réseau expérimental.

Pour répondre aux objectifs de fiabilité, de flexibilité du contrôle et d'autonomie des différents processeurs, le système DFS a une structure très modulaire. De plus les modules relativement indépendants qui

composent le système DFS sont répartis sur plusieurs processeurs distincts. Ainsi au prix d'un minimum de redondance, la défaillance de certains modules (FARBER et al. RES-17) ne doit avoir que des conséquences très limitées, toute assistance nécessaire à un processeur pouvant venir de plusieurs sources différentes.

Pour des raisons de fiabilité de l'ensemble du système de gestion de fichiers, le mode de désignation a une structure arborescente redondante, chaque niveau de la hiérarchie contenant l'intégralité des informations contenues dans les niveaux supérieurs.

Le catalogue général est dupliqué en n copies identiques (n étant le nombre de processeurs).

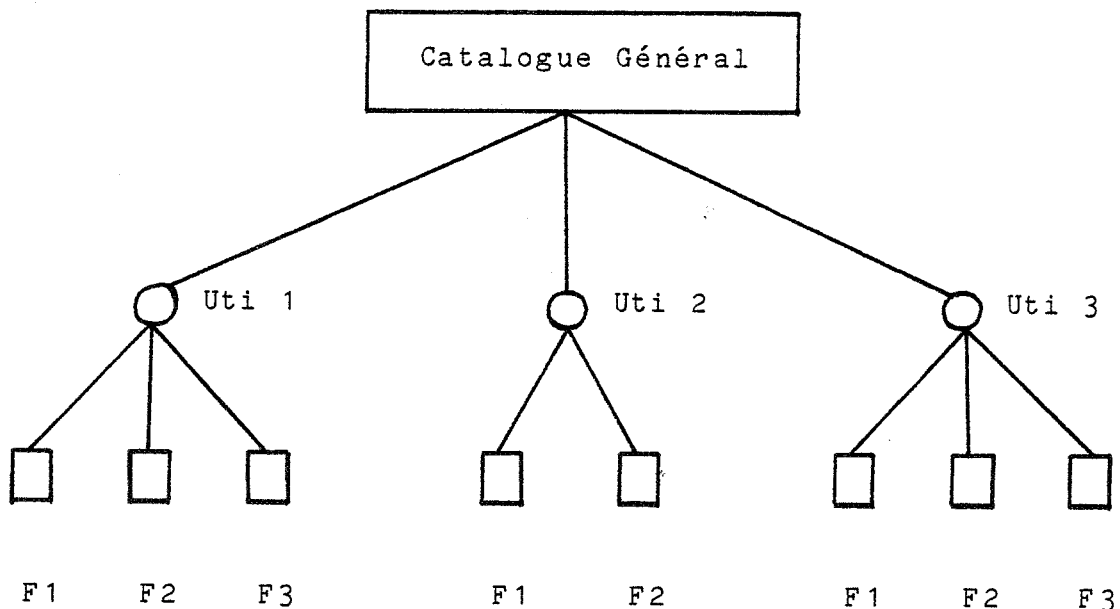


Figure - 2. Structure de désignation de DFS.

Le contrôle est réparti et sur chaque processeur il y a un processus assurant à la fois :

- l'allocation d'espace disque,
- la création de fichiers,
- leur effacement.

Il est à noter que sur ce système la localisation d'un fichier déjà existant se fait en envoyant un message à tous les processus racines chargés de gérer les catalogues généraux. Seul le site abritant le fichier répond en spécifiant le catalogue utilisateur.

L'accès proprement dit se fait grâce à un processus d'accès ("access process") qui est créé par le système d'exploitation abritant le fichier.

Pour la création d'un nouveau fichier, il faut d'abord faire une demande d'allocation de ressources. Le dialogue se déroule ensuite suivant le schéma précédent, c'est à dire via un processus d'accès. Le processus d'accès est un véritable canal par où transitent toutes les requêtes d'accès au fichier. Grâce à cette facilité les fichiers sont adressés comme des processus, ce qui accroît leur souplesse d'utilisation.

Quant au problème de partage de données, il est réglé au niveau des accès. Ainsi au lieu d'avoir des données dupliquées, deux ou plusieurs références distinctes peuvent pointer le même fichier qui se trouve ainsi partagé par plusieurs utilisateurs.

3. Méthode d'accès direct réseau MADRE

3.1. Environnement de MADRE

Le but de MADRE est d'offrir l'accès à un ensemble d'informations réparties sur un réseau général hétérogène en respectant les objectifs suivants (CHUPIN, SEGUIN: GFR-2) :

- accès à des fichiers en organisation directe,
- hétérogénéité des machines participant au réseau,
- définition d'un protocole de fichier minimum,
- standardisation des mécanismes de communication;
MADRE utilise pour cela l'architecture existante d'un service de transport.

3.2. Fonctionnalité de MADRE

Conçu pour répondre aux impératifs de nombreuses mises à jour interactives, MADRE est une application orientée vers la répartition d'une méthode d'accès. L'approche client-serveur a été choisie. Chaque site participant dispose d'un sous-système MADRE composé d'un client et d'un serveur.

L'utilisateur s'adresse à la partie client qui joue le rôle d'un multiplexeur et interpréteur de requêtes. La partie "serveur" établit la correspondance entre les requêtes MADRE et les ordres du système de gestion de fichiers local.

Les fichiers accessibles du réseau sont des fichiers à accès direct et ils sont gérés par un répertoire réseau dont une copie est présente sur chaque site. Chaque article du répertoire réseau est constitué d'un certain nombre de champs comme l'indique la figure suivante.

NMFR	NBPAR	ADM	SITE	NMSGF	OPLM	BLKSZ	MODE
nom du fich. réseau	nombre de parti- tions	nom admi- nistra- tif	site de rési- dence	nom sgf far	opl volume rési- dence	taille du bloc	mode ex- ploi- tation

far : fichier accessible du réseau

opl : label opérationnel

Figure - 3. Répertoire réseau de MADRE.

3.3. Mise en oeuvre de MADRE

Madre met à la disposition de l'utilisateur un certain nombre de primitives faisant appel à des modules qui sont insérés dans le programme utilisateur au moment de l'édition de liens.

Les principales primitives sont:

- des primitives d'accès,
- des primitives d'utilisation,
- des primitives d'acquiescement et de test de l'état des requêtes.

4. Protocole d'accès aux données (Data Access Protocol) (DEC: GFR-3)

DAP est un protocole d'accès aux données distantes qui opère à un niveau utilisateur et il offre une véritable méthode d'accès réseau.

Ce protocole s'intègre dans l'architecture DNA ("Digital Network Architecture") qui est un ensemble hiérarchisé de protocoles englobant les niveaux fonctionnels suivants:

- niveau lien physique,
- niveau lien logique,
- niveau dialogue.

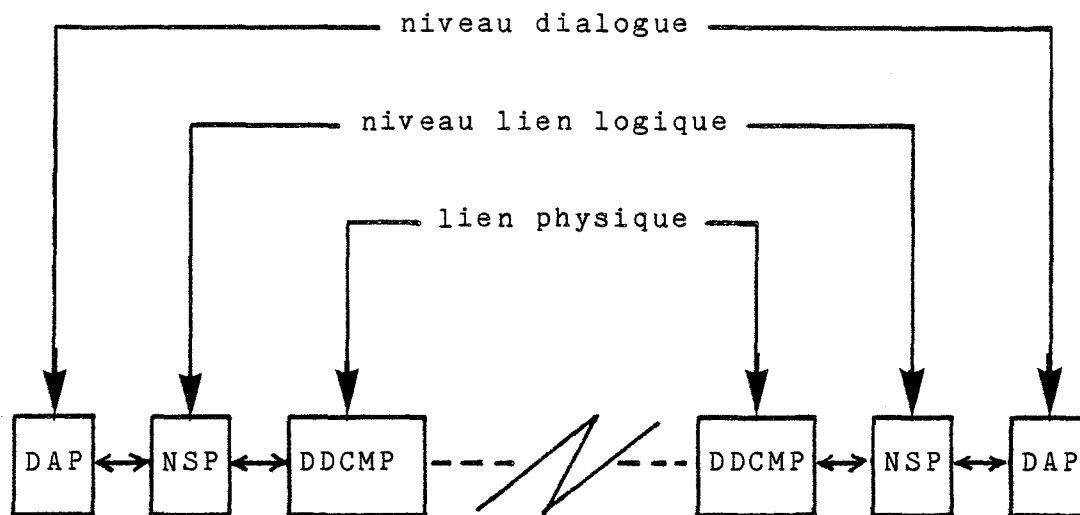


Figure - 4. Hiérarchie de niveaux dans DNA.

4.1 Les différents niveaux de DAP

1) Le niveau lien physique:

Il est constitué d'un protocole de communication le DDCMP (Digital Data Communication Protocol) opérant en half ou full-duplex, en mode synchrone, asynchrone ou circuits parallèles.

2) Le niveau lien logique:

Il comprend un protocole, le NSP (Network Service Protocol), permettant d'établir un lien logique entre deux processus distants. En plus de la communication inter-processus, ce niveau fonctionnel offre aussi un routage des messages entre les différents noeuds.

3) Le niveau dialogue:

C'est ce niveau qui nous intéresse particulièrement ici, car DAP est le protocole d'accès aux données.

4.2. Description fonctionnelle de DAP

DAP apparaît au NSP comme un programme utilisateur. Compte tenu de son environnement ce protocole offre une interface avec chacune des entités suivantes:

- le SGF local,
- les programmes utilisateurs,
- le NSP qui joue le rôle d'une machine de transport.

Chaque système d'exploitation, compatible avec DAP, offre des facilités de traduction de ses structures internes d'entrée-sortie en structures standards DAP et inversement.

En plus des possibilités de transfert de fichiers qui sont offertes, DAP est une véritable méthode d'accès réseau.

Les primitives d'accès distant que procure DAP incluent des ordres comme READ, WRITE, OPEN, CLOSE, DELETE.

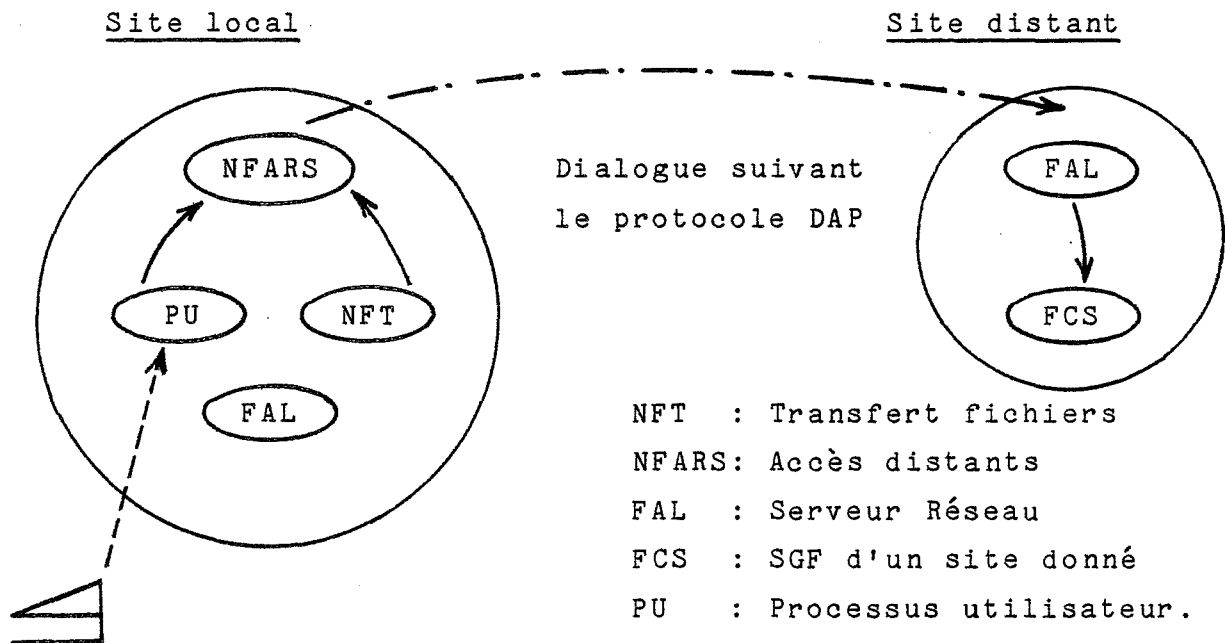


Figure - 5. Les composants fonctionnels de DAP.

5. Le système DISCO (Distributed Database for Small Computers) (HOLLER: GFR-6)

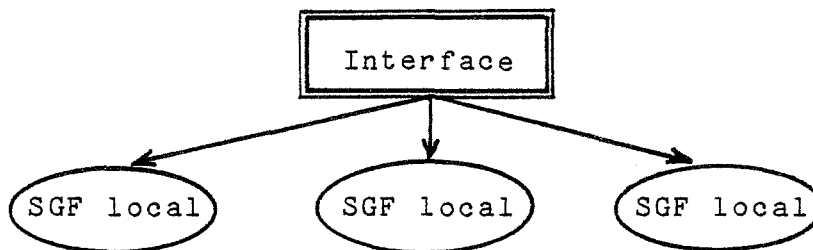
5.1. Généralités et but du système DISCO

Le système DISCO est construit sur un réseau hétérogène de mini-ordinateurs. Ses caractéristiques principales sont les suivantes :

- le système, décomposé en modules qui peuvent être répartis et/ou dupliqués, gère des fichiers partitionnés,
- les articles nécessaires à une application appartiennent à la même partition,
- la partition est localisée sur le même site que l'application.

5.2. Niveau de l'interface DISCO

En se référant au modèle ANSI/SPARC/X3 (cf. chapitre IV "Modèles d'architecture"), on voit que dans le système DISCO, l'interface chargée de faire l'aiguillage est située au dessus des différents systèmes de gestion de fichiers locaux. Ceux-ci sont considérés comme partie intégrante des systèmes d'exploitation.



5.3. Allocation de ressources distribuées

L'allocation décentralisée de ressources dans DISCO est réalisée au moyen de contrôleurs répartis communiquant suivant des protocoles spécialisés de haut niveau. Ces protocoles, fondés sur la communication inter-processus, permettent aux contrôleurs de se mettre d'accord sur la requête à satisfaire.

Deux types de protocoles de coordination ont été étudiés. Ils correspondent :

- soit à des systèmes de contrôleurs fortement couplés où chaque contrôleur a une copie de l'état global d'allocation des ressources réparties,

- soit à des systèmes faiblement couplés où chaque contrôleur ne connaît que l'état d'allocation des ressources qui lui sont locales . Une détection d'interblocages ainsi que des procédures de reprise sont alors nécessaires.

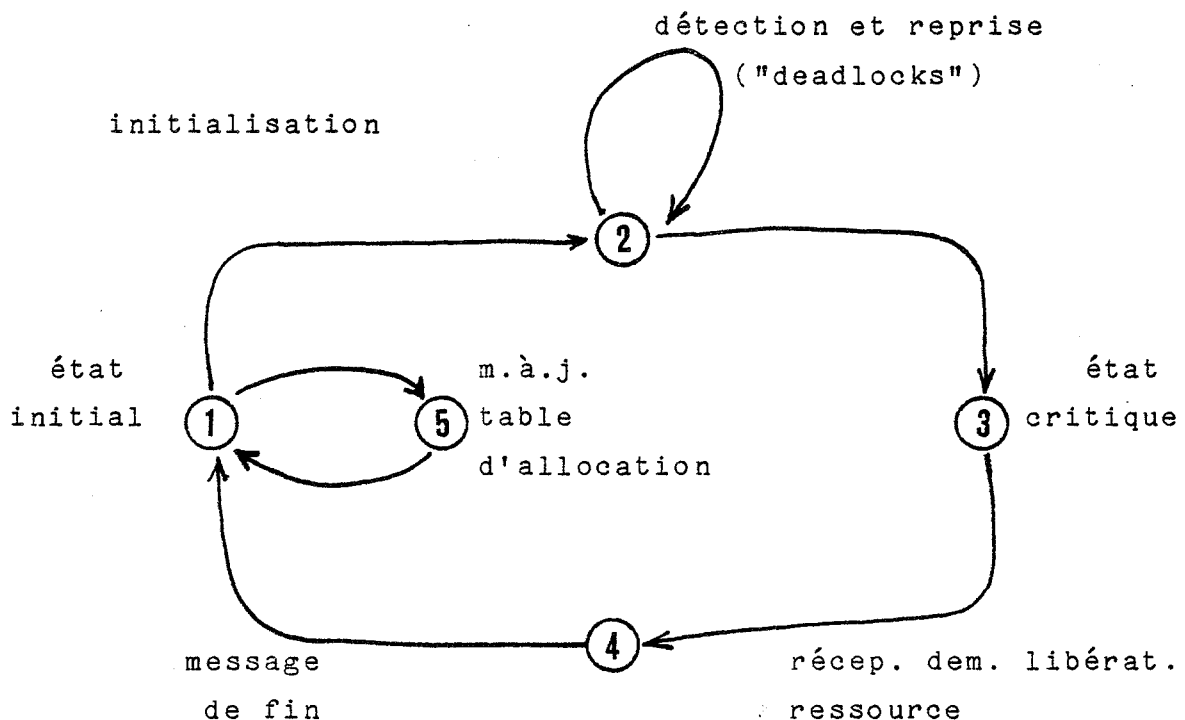


Figure - 6. Protocole de coordination
(système faiblement couplé).

5.4. Gestion du catalogue des fichiers logiques

Le but de cette gestion est d'offrir un unique espace de désignation répondant aux critères suivants :

- indépendance par rapport au noeud,
- exploitation du parallélisme,

- flexibilité et disponibilité,
- interface minimale et facilité d'implémentation.

Deux catégories de fichiers existent dans le système :

- les fichiers contenant des données ; ils sont répartis par région (le critère de répartition est fondé sur l'appartenance à un groupe d'utilisateurs dont on connaît la répartition),
- les fichiers contenant les chemins d'accès.

CHAPITRE III

LES SYSTEMES DE GESTION DE BASES DE DONNEES REPARTIS

1. Diverses approches de la distribution
 - 1.1. Méthode descendante ou répartition
 - 1.2. Méthode ascendante ou coopération
2. Analyse de quelques SGBDR
 - 2.1. L'architecture de SDD-1
 - 2.2. Le modèle ADD
 - 2.3. Le projet FRERES
 - 2.4. Le projet SIRIUS-DELTA

1. Les diverses approches du problème de la distribution

A l'heure actuelle il existe plusieurs Systèmes de Gestion de Bases de Données (SGBD) fournis par les constructeurs: IMS, SOCRATE, SYSTEM-R, TOTAL, etc. Chacun de ces systèmes a son modèle de structuration des informations qui lui est propre (modèle hiérarchique, modèle réseau, modèle relationnel). Ces systèmes hétérogènes ne peuvent pas communiquer entre eux et ils sont donc utilisés indépendamment les uns des autres.

Ainsi à côté des utilisateurs ayant accès déjà à diverses bases de données en exploitation dans différents sites géographiques et désirant simplifier leur utilisation en les faisant coopérer, on en trouve d'autres voulant concevoir, pour des buts spécifiques et bien définis, des bases de données décentralisées voire réparties.

On se trouve donc en face de deux situations bien distinctes dont l'une est caractérisée par la préexistence de bases de données locales déjà en exploitation et l'autre par un champ presque inexploré où tout est à faire.

Il apparaît donc que le problème de la distribution dans les bases de données se pose en fait de deux façons différentes au moins. Ces deux manières induisent évidemment deux méthodes de résolution :

- une méthode descendante pour une répartition,
- une méthode ascendante pour une coopération de bases existantes.

Nous étudierons par la suite ces deux méthodes tout en sachant que les concepteurs n'utilisent pas exclusivement l'une ou l'autre et que les frontières entre la répartition et la coopération dans les bases de données ne sont pas bien nettes. Un système peut apparaître, par certains de ses aspects fonctionnels, plutôt comme un système réparti, le même système vu sous l'angle de ses composants répartis (qui sont différents sous-systèmes autonomes) fera penser à une coopération. C'est pourquoi le problème du niveau fonctionnel d'une application est très important.

1.1. Méthode descendante ou répartition

Elle peut consister à partir d'une base de données centralisée et à la répartir sur différents sites. Elle peut consister aussi à concevoir de toutes pièces une base de données répartie.

La répartition concerne le stockage des données, les traitements, et les fonctions d'accès.

1.1.1. Exemple du projet ETOILE (MEYER: BDR-14)

Un exemple de cette approche est fournie par l'application développée à l'Université de Nancy sur la gestion financière et comptable du CNRS¹. Entrant dans le cadre du projet SIRIUS² sur les bases de données réparties, cette étude se propose de répartir une base de données qui à l'origine était locale.

(1) CNRS : Centre National de la Recherche Scientifique et Technique.

(2) SIRIUS : Projet Pilote sur les Bases de Données Réparties.

Le modèle de répartition utilisé fixe comme objectifs l'obtention de plusieurs bases locales reliées entre elles par un "logiciel de répartition et de cohérence".

En effet, le fonctionnement réparti de l'application nécessite la présence sur chaque site d'un logiciel chargé d'assurer toutes les fonctions de répartition et de cohérence. C'est l'ensemble de ces logiciels qui constitue une interface entre les différentes bases locales et qui permet d'avoir une vue globale de la base répartie.

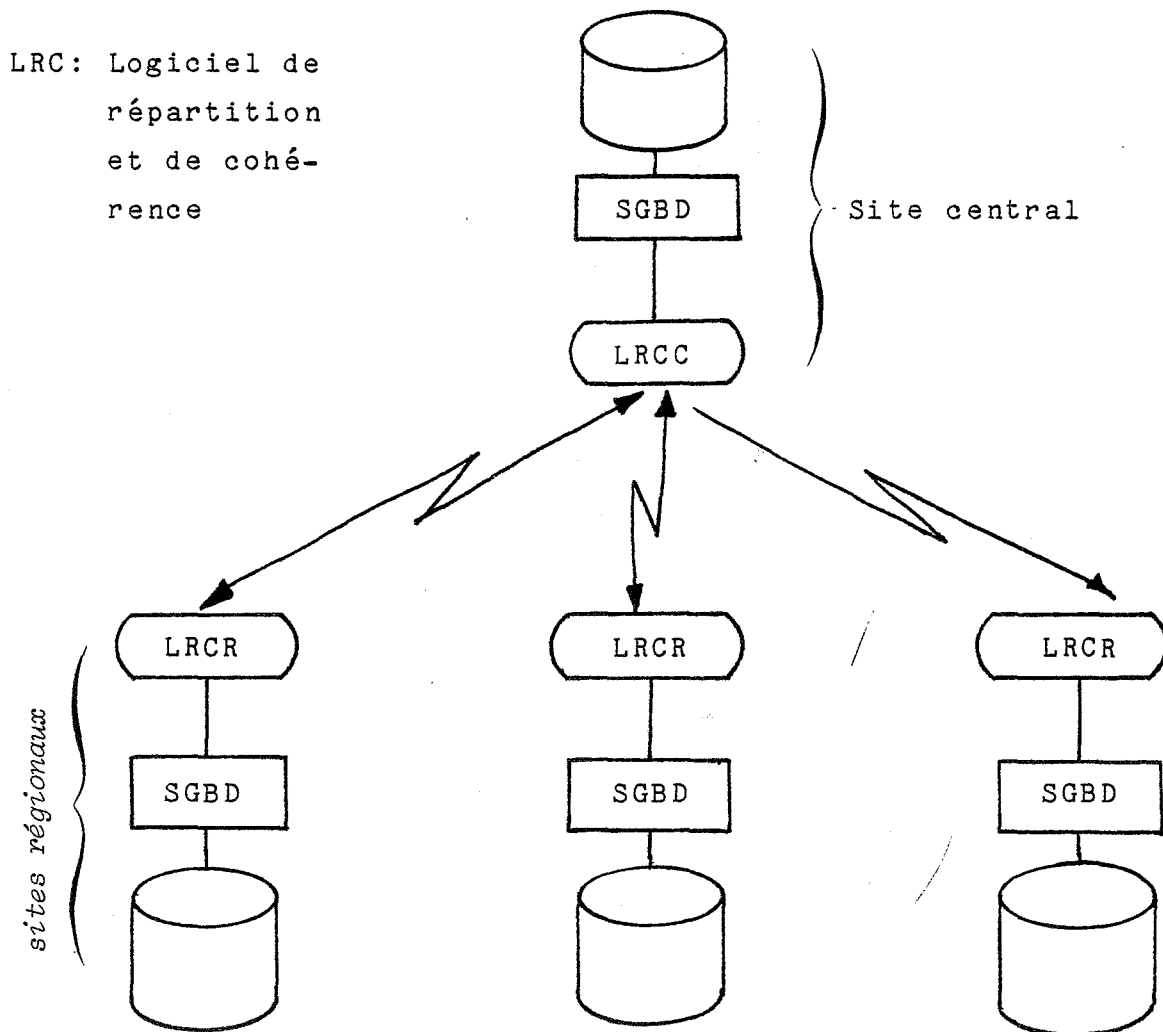


Figure - 1. Architecture globale du système ETOILE

Les sites locaux sont autonomes dans la mesure où chacun dispose de l'intégralité des données qui lui sont

nécessaires pour les traitements locaux et les consultations locales. D'où une certaine redondance dans les données.

Les terminaux sont gérés en local par chaque site et ainsi l'utilisateur n'accède à la base qu'en local. Même si les mises à jour effectuées ont des répercussions sur les autres composantes de la base de données répartie, ces répercussions se font de manière automatique et entièrement transparente à l'utilisateur.

Cependant la base centrale et les bases régionales doivent tout de même avoir "la même description logique". De plus, l'existence d'un catalogue central permettant la localisation des données fait de la partie centrale un point névralgique. Donc les défaillances du composant central sont assez critiques pour la base répartie toute entière.

Les principales contraintes imposées par ce système se situent au niveau des liens sémantiques qui doivent unir les différentes bases régionales. Ces liens sont assurés par la même description logique des bases. L'autre contrainte principale est l'absence de symétrie qui oblige à centraliser le contrôle. De plus la cohérence des bases locales entre elles n'est pas requise à tout instant : on se contente de les remettre régulièrement en "phase".

1.2. Méthode ascendante ou coopération

Elle intéresse ceux qui disposent déjà de bases de données opérationnelles mais exploitées indépendamment les unes des autres. Ces bases, pour la plupart hétérogènes, ont pourtant certains liens sémantiques: les informations traitées ne sont pas indépendantes.

C'est pourquoi ceux qui exploitent ces bases voudraient bien voir coopérer les différents Systèmes de Gestion de Bases de Données (SGBD). Cette coopération exige que les différents SGBD s'entendent sur un minimum de conventions ne serait-ce qu'au niveau conceptuel. En effet, faire coopérer différentes entités implique en général la définition d'une vue commune de ces entités qui est en principe fondée sur leurs différentes caractéristiques.

Ainsi un premier niveau de coopération entre SGBD pourrait être fondé sur le principe que "faire coopérer deux SGBD, c'est étudier les possibilités de combinaisons entre les éléments des deux SGBD de manière à accroître les possibilités de l'un des deux SGBD, tout en conservant à chacun son intégrité propre et son autonomie" (BRUNET, LEPAPE: BDR-5)

En règle générale, la plupart des études effectuées sur la coopération de bases de données posent le principe de la définition d'une vue commune aux différentes bases de données. Ainsi, utiliser une base de données répartie, c'est en premier lieu (SPACCAPIETRA: BDR-16) pouvoir en constituer une description globale qui permettra au SGBDR de fournir cet accès intégré à l'ensemble des données réparties que vont demander les utilisateurs.

1.2.1. La coopération dans POLYPHEME (ADIBA: BDR-1)

POLYPHEME part de l'existence préalable de bases locales pouvant être hétérogènes. C'est pour cela que le modèle relationnel s'est avéré plus judicieux compte tenu de la grande hétérogénéité des SGBD participants et des

possibilités d'évolution ultérieure. La coopération dans POLYPHEME se situe en effet à un niveau global tandis que chaque site reste maître de sa base. Le noyau de POLYPHEME est constitué d'un logiciel chargé de gérer la coopération.

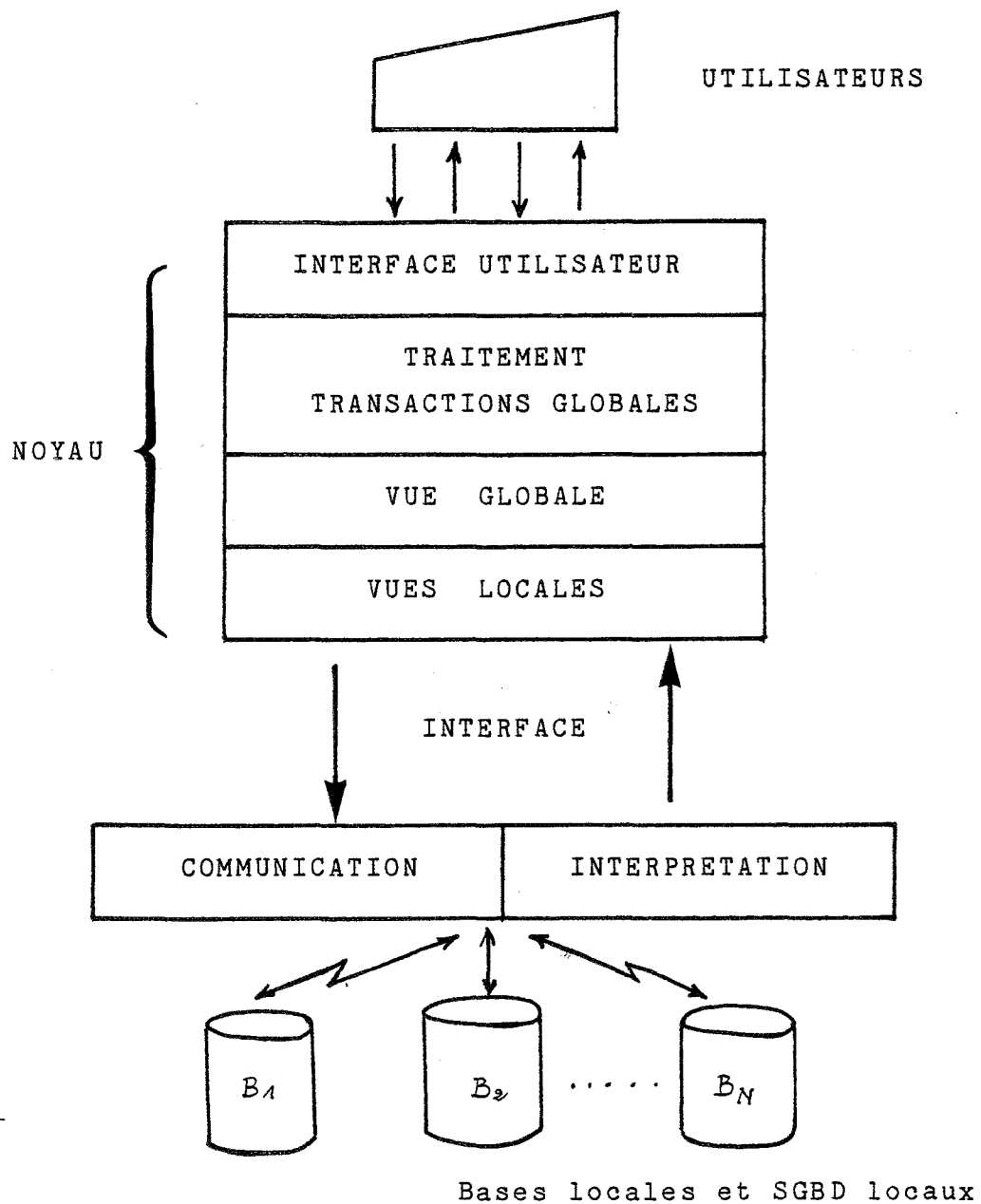


Figure - 2. Schéma fonctionnel de POLYPHEME

Pour ce faire, il dispose d'une vue globale des bases locales coopérantes. Il prend en compte les transactions globales et les analyse. Il possède en outre un mécanisme de décomposition des transactions globales en transactions locales. Les SGBD locaux communiquent avec le noyau grâce à une interface chargée de gérer plus spécifiquement la communication. Cette communication utilise bien sûr le réseau qui est en l'occurrence le réseau CYCLADES (POUZIN: RES-31)

La partie interprétation complète cette architecture de POLYPHEME. Les informations circulant entre le noyau et les SGBD locaux sont transformées par ce niveau fonctionnel en fonction de leur origine et de leur destination.

Cependant, il faut souligner que la première version opérationnelle de POLYPHEME traite uniquement des bases de données homogènes.

2. Analyse de quelques SGBDR

Les nouveaux problèmes posés par les Systèmes de Gestion de Bases de Données Réparties ont suscité beaucoup d'intérêt chez de nombreuses équipes de recherche.

C'est ainsi qu'en France, dès 1976 a été lancé le Projet Pilote SIRIUS (LE BIHAN et al. BDR-13) dont le premier objectif fut d'organiser et de coordonner au niveau national les efforts qui étaient faits pour acquérir la maîtrise de ce nouveau savoir faire. Il a permis, dans sa première phase, la réalisation de plusieurs prototypes (ETOILE, FRERES, POLYPHEME ...).

Dans sa deuxième phase, SIRIUS devait permettre de faire la synthèse des travaux menés dans le cadre du projet - au cours de la phase exploratoire - par la réalisation d'un prototype industriel SIRIUS-DELTA autorisant une intégration plus globale des problèmes posés par les SGBDR.

Pendant ce temps :

- en Allemagne des projets comme DISCO (HOLLER: GFR-6), ESA, POREL et VDN ;
- au Canada, le projet ADD (TOTH et al. BDR-19);
- aux Etats Unis, les projets "distributed INGRES" (STONEBRAKER, NEUHOLD: BDR-17), SDD-1 (ROTHNIE: BDR-15)

ont vu le jour.

2.1. Architecture de SDD-1 (System for Distributed Database) (ROTHNIE: BDR-15)

L'architecture de SDD-1 repose sur la coopération de sous-systèmes fonctionnels bâtis autour de SGBD locaux homogènes ayant un interface de transport avec le réseau ARPA.

Cette coopération utilise le modèle relationnel (CODD: MAS-5). Elle est organisée par un sous-système composé de modules géographiquement distribués qui ont pour rôle :

- d'une part de traduire les requêtes des usagers en un certain nombre de processus locaux,

- d'autre part de coordonner l'exécution répartie qui en résulte.

L'unité de répartition est le fragment de relation. L'optimisation du temps de réponse des requêtes est fondée sur la réduction du nombre de transferts réseaux. Elle utilise le principe de l'évaluation statique en début de requête.

La synchronisation des mises à jour est faite en datant les transactions et les copies de fichiers. Quant aux mises à jour elles-mêmes, elles sont mémorisées temporairement dans des fichiers différentiels. En fin de transaction, une écriture de la base est faite suivant le mécanisme de la consolidation à deux phases ("two step commitment").

2.2. Le modèle ADD (Architecture for Distributed Databases) (TOTH et al. BDR-19)

L'architecture du système ADD est formée d'un ensemble hiérarchisé de sept niveaux. Chaque niveau est associé à un processeur (logiciel) qui effectue :

- soit les transformations adéquates sur les requêtes,
- soit les opérations que nécessitent celles-ci.

Les quatre couches supérieures sont identiques sur tous les sites. Deux couches dans cette architecture sont plus spécifiquement orientées vers la répartition :

- le schéma de partitionnement qui définit une décomposition logique des relations de la vue globale en sous-relations ou partitions disjointes;
- le schéma de répartition qui définit la localisation géographique des segments ou partitions sur les différents sites du réseau.

La couche de niveau quatre (associée au schéma de répartition) offre un interface aux différentes vues locales des SGBD participants. Elle assure aussi le routage des actions vers les bases locales concernées. Le processeur associé au sous-schéma local transforme les requêtes relationnelles en ordre d'interrogation et de mises à jour compatibles avec les chemins d'accès locaux. Ceux-ci sont bâtis au dessus des Systèmes de Gestion de Fichiers.

A P P L I C A T I O N S U T I L I S A T E U R S

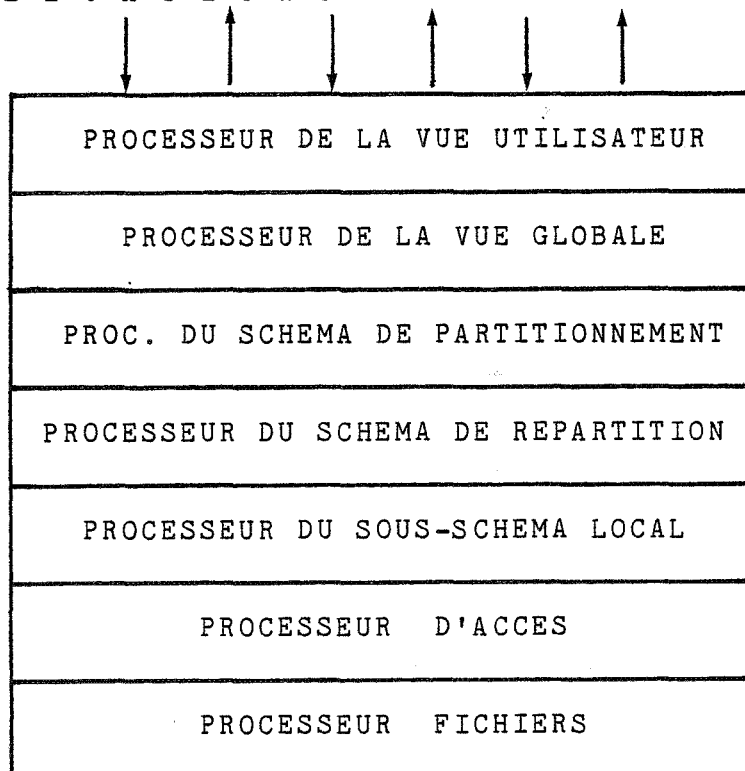


Figure - 3. Les niveaux du modèle ADD

2.3. Le projet FRERES (BOSC: BDR-3), (BOSC, CHAUFFAUT: BDR-4)

2.3.1. Caractéristiques

Cette application est née en tant que Système de Gestion de Fichiers Répartis puis elle a été étendue par la suite à la gestion et surtout à l'interrogation de collections de données (bases de données, fichiers classiques).

Les principales caractéristiques du système FRERES sont :

- FRERES est un serveur mono-utilisateur,
- les bases locales formant la BDR décrivent des objets de même nature (collections de données sémantiquement voisines) ; cependant elles peuvent être hétérogènes,
- le système comprend un site maître et des sous-systèmes esclaves.

2.3.2. Architecture fonctionnelle

1) Fonctions du système d'interrogation

L'interrogation a pour but d'obtenir des informations répondant à un critère donné. Ces informations doivent être présentées sous forme d'un fichier de résultats homogènes. Comme les bases peuvent être hétérogènes, il est nécessaire d'assurer une conversion de fichiers selon l'un des mécanismes suivant :

- un mécanisme général utilisant un langage de description de structure,
- un mécanisme particulier à chaque collection de données.

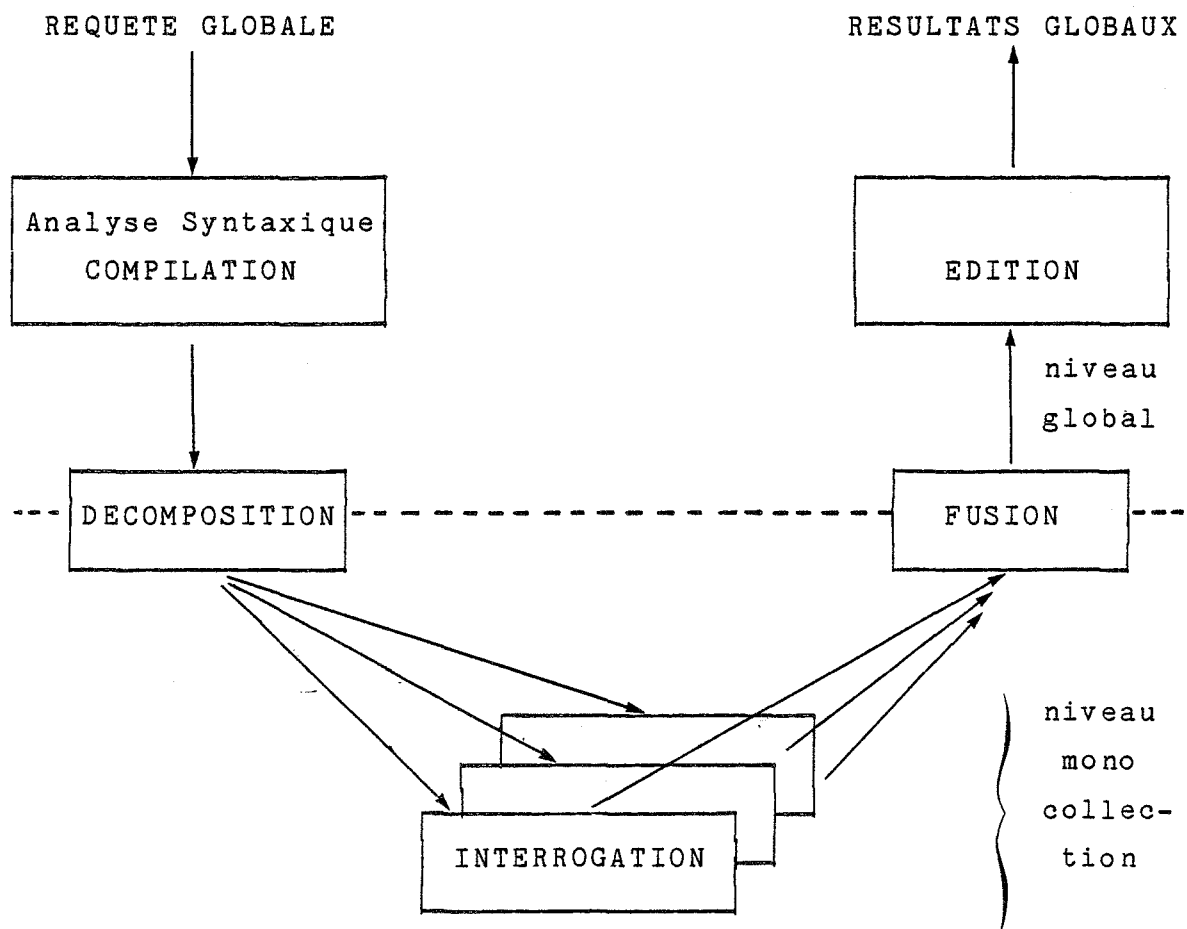


Figure - 4. Fonctions du système FRERES

2) Fonctions de répartition

L'exécution d'une requête sur le système FRERES nécessite trois phases principales :

- une phase d'analyse syntaxique-compilation pendant laquelle la requête globale est traduite en

un langage intermédiaire (unique) où la manipulation des données est exprimée indépendamment de leur localisation;

- une phase de décomposition et ordonnancement produisant un plan d'exécution répartie.

- une phase d'exécution répartie proprement dite.

A ces différentes phases correspondent trois grandes catégories de fonctions :

- les fonctions qui assurent le traitement de la question au niveau global (analyse, compilation) et le dialogue avec l'utilisateur puis à la fin du traitement l'édition des résultats. Ces fonctions sont regroupées sur le site maître. Ce dernier abrite aussi le catalogue associé à la vue globale (celle-ci donne une description unique des données).

- les fonctions qui exécutent une requête mono-collection sur un site (adaptation à la vue locale consistant à introduire les fonctions d'accès associées aux articles logiques ; exécution puis conversion c'est-à-dire structuration des résultats selon le format global) ;

- la fonction d'exécution du plan qui

- * envoie les requêtes aux divers systèmes concernés ainsi que les domaine et échelle de correspondance nécessaires à la conversion,

- * prend en charge la synchronisation des réponses.

Chaque calculateur esclave abrite un catalogue associé à chaque vue locale ainsi que les fonctions traitant les requêtes sur un site.

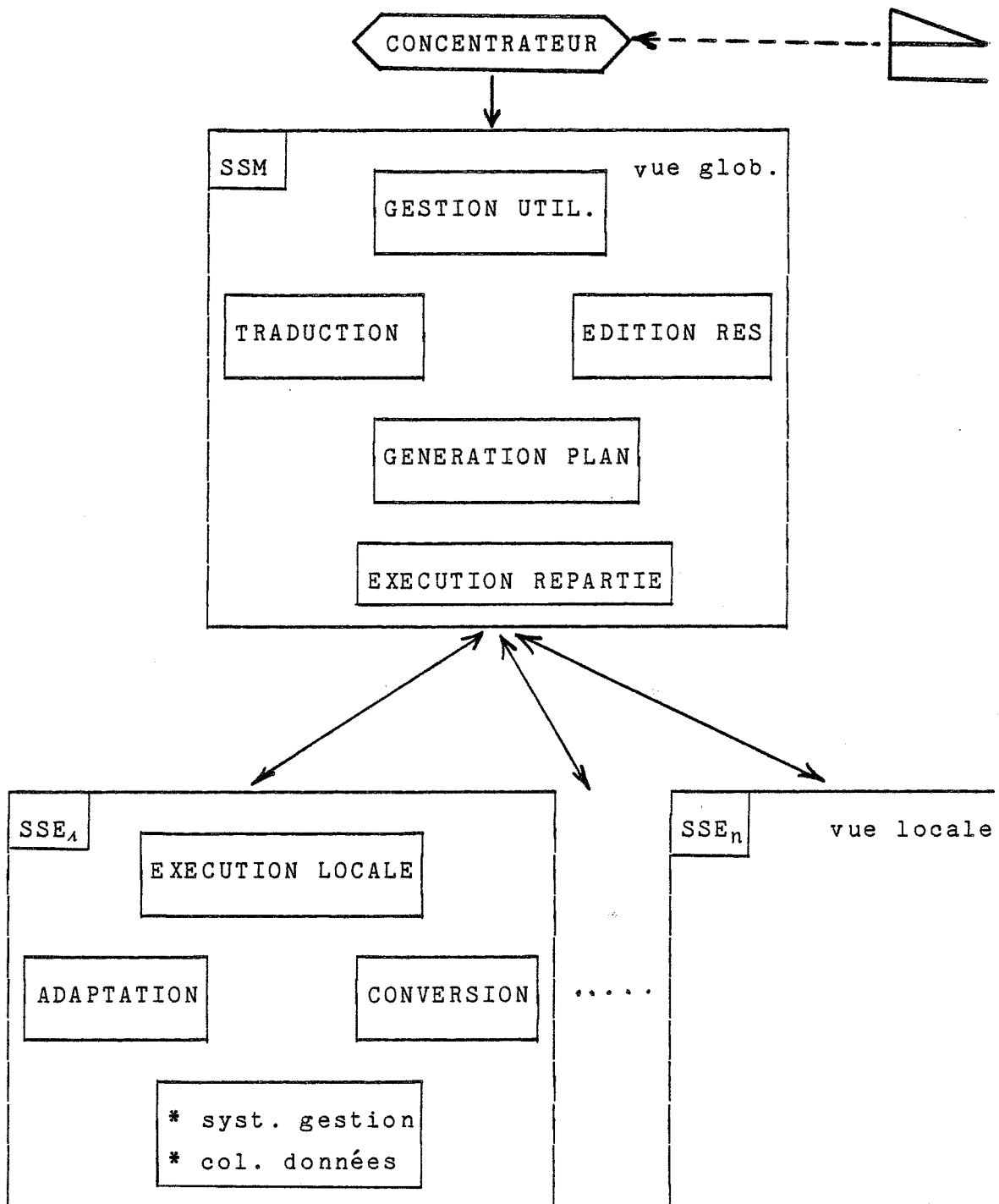


Figure - 5. Schéma de répartition

2.4. SIRIUS-DELTA (LE BIHAN et al BDR-13)

2.4.1. Objectifs

SIRIUS-DELTA a pour objectifs la définition et la réalisation d'un Système de Gestion de Bases de Données Réparties (SGBDR) sur un réseau local de mini-ordinateurs.

2.4.2. Fonctionnalité

L'architecture en couches de SIRIUS-DELTA repose sur quatre niveaux fonctionnels :

- fonction de gestion classique de Bases de Données (interprétation de requêtes, accès aux données locales) utilisant un SGBD développé par un constructeur de mini-ordinateurs;
- fonction de gestion de données spécifique au contexte réparti (décomposition de requêtes en fonction de la localisation);
- fonction de contrôle réparti;
- fonction d'exécution répartie (activation de processus répartis, transfert de données, enchaînement de programmes suivant un plan d'exécution répartie, synchronisation de processus répartis.

2.4.3. Architecture de SIRIUS-DELTA

SIRIUS-DELTA intègre plusieurs couches dont chacune constitue un sous-système assurant une fonction déterminée.

1) SER : Système d'Exécution Répartie

Le fonctionnement réparti de SER utilise trois entités de base :

- l'action locale : unité d'exécution logique correspondant physiquement à l'exécution d'un programme sur un processeur (consultation, fusion),
- les variables de synchronisation permettant de gérer les enchaînements d'actions locales, leurs conditions d'activation et leur synchronisation,
- les fichiers de données temporaires servant de moyen d'échange de données entre une action locale productrice et l'action locale consommatrice.

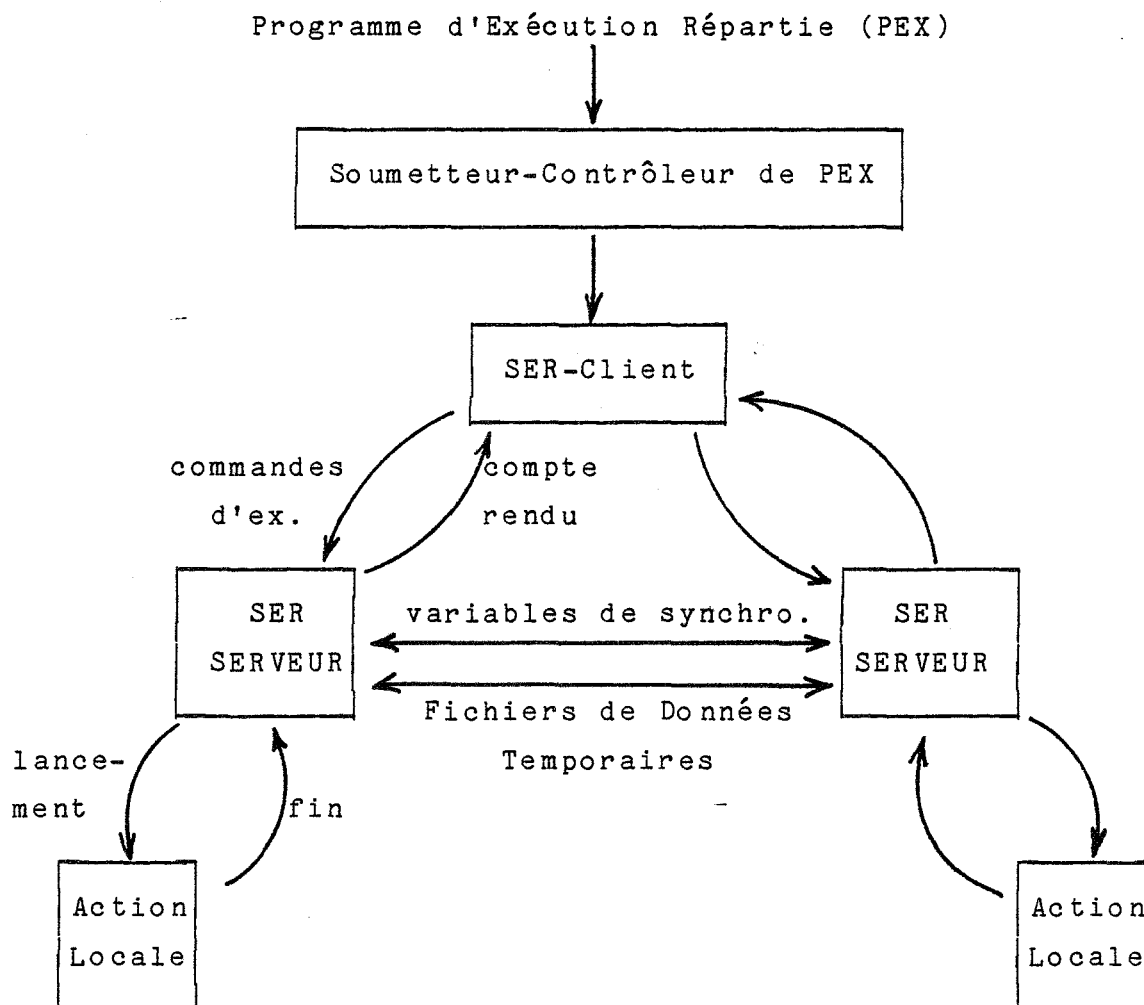


Figure - 6. Structure d'exécution répartie

2) SCORE : Système de Contrôle Réparti

Ce système basé sur l'utilisation de techniques d'anneau virtuel et de séquenceur circulant (LE LANN: CAC-8, CAC-9) fournit les mécanismes nécessaires au maintien de la cohérence de la base de données répartie. Celle-ci est structurée en objets logiques accessibles individuellement.

La fonction essentielle de SCORE est d'assurer un ordonnancement cohérent de l'ensemble des demandes d'accès dans un système soumis à des défaillances. SCORE utilise un protocole de validation à deux phases.

3) SILOE : Système d'interprétation, de Localisation et d'Evaluation

Ce système assure la localisation des données, l'évaluation répartie des requêtes et la génération de programmes d'exécution répartie (PEX).

Le niveau SILOE local assure aussi l'interface avec les SGBD locaux.

CHAPITRE IV

MODELES D'ARCHITECTURE DE SYSTEMES

1. Nécessité d'un modèle commun d'architecture
2. Le modèle ANSI/SPARC/X3
3. Le modèle de référence de l'ISO
 - 3.1. Objectifs
 - 3.2. L'architecture à sept couches
4. Architecture orthogonale
5. Techniques de structuration
 - 5.1. Structuration par emballage
 - 5.2. Structuration en couches
 - 5.2.1. Principe de la structuration
 - 5.2.2. Problème de communication
6. Quelques approches de la répartition de données
 - 6.1. L'approche séparatrice
 - 6.2. L'approche unificatrice
 - 6.3. Notre approche : un palliatif

1. Nécessité d'un modèle commun d'architecture

L'étude des différents systèmes de distribution de données permet de mettre en évidence deux constatations majeures:

- d'une part, si l'objectif initial commun des concepteurs est d'offrir aux utilisateurs la possibilité d'accéder à des données non locales et celle de stocker ces données là où elles sont produites ou bien là où elles sont utilisées, les moyens pour y parvenir diffèrent beaucoup;
- d'autre part, en l'absence de normes communes et unanimement respectées, les concepteurs de systèmes répartis apportent parfois des solutions originales, mais assez souvent trop liées à l'environnement, à des problèmes pourtant assez généraux. Ces solutions très particulières, fondées sur les spécificités des applications concernées, ne sont évidemment pas transportables.

De ces deux constatations il résulte un embarras pour tout nouveau concepteur d'une application répartie au milieu d'une multitude de réalisations où il n'est pas très aisé de dégager les caractéristiques générales.

C'est pour ces raisons, qu'il nous a paru primordial d'articuler notre démarche autour du modèle de référence pour l'interconnexion de systèmes ouverts de l'ISO dont nous rappelons plus loin les grandes lignes (§ 3).

2. Le modèle ANSI (ANSI/SPARC/X3: MAS-2)

Le modèle ANSI/SPARC est composé d'une architecture à quatre niveaux. Il est particulièrement orienté vers les Systèmes de Gestion de Bases de Données.

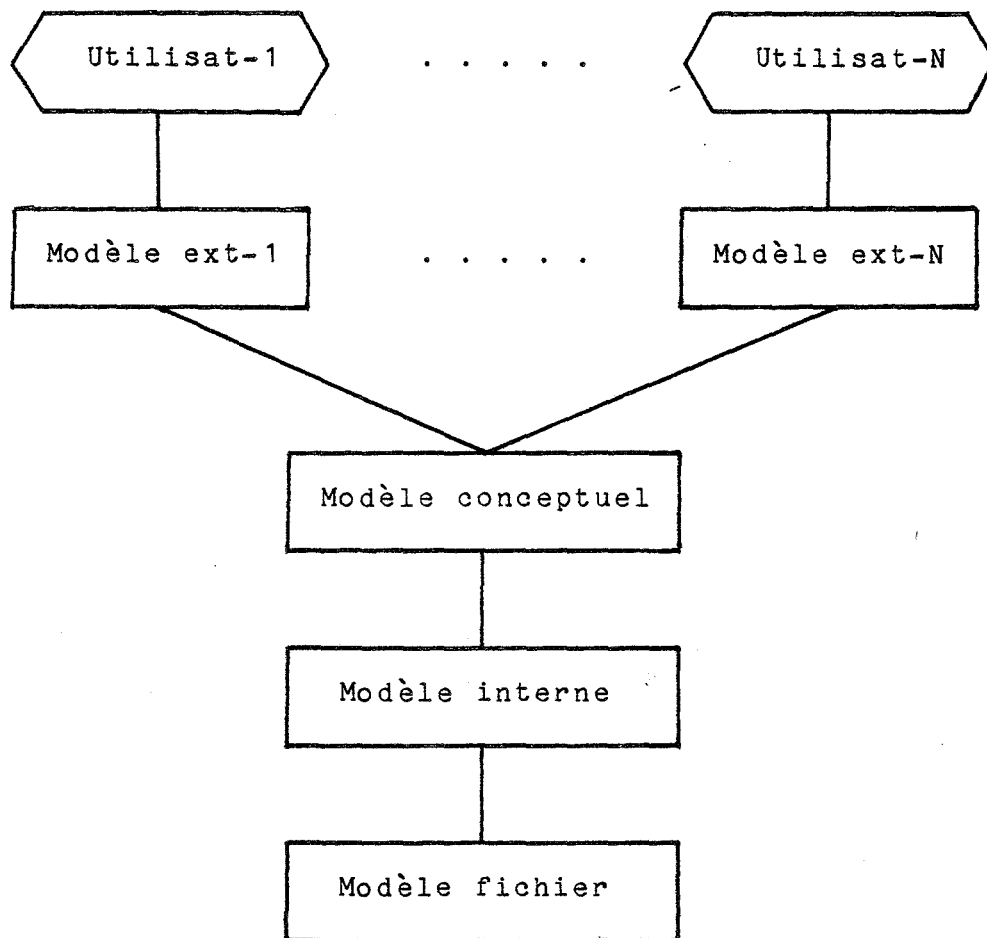


Figure - 1. Modèle ANSI/SPARC

1) Modèle externe : il modélise la base de données suivant la vision d'utilisateurs individuels ou de groupes d'utilisateurs. C'est un modèle simplifié du monde réel tel qu'il est vu par une ou plusieurs applications.

2) Modèle conceptuel

Il offre une vision globale du monde réel prise en compte par le SGBD et une formalisation non ambiguë des informations. Ce modèle est construit et géré pour l'ensemble des applications.

3) Modèle interne

Il contient les outils qui décrivent les données stockées et les moyens d'accéder aux données.

4) Modèle fichier

Il décrit la base de données comme un ensemble de fichiers logiques pouvant être stockés sur différents périphériques en utilisant différents modes d'organisation.

On peut dire que déjà avec le groupe de travail DBTG ("Data Base Task Group de Codasyl") l'idée fondamentale du rapport CODASYL avait été de séparer la description globale de la base de données (le schéma) des descriptions multiples correspondant à des utilisations particulières de la base de données (les sous-schémas).

Le système DISCO part de ce modèle d'architecture et étudie différentes alternatives pour l'emplacement du mécanisme chargé de faire l'aiguillage entre les requêtes de l'utilisateur et les sous-systèmes locaux.

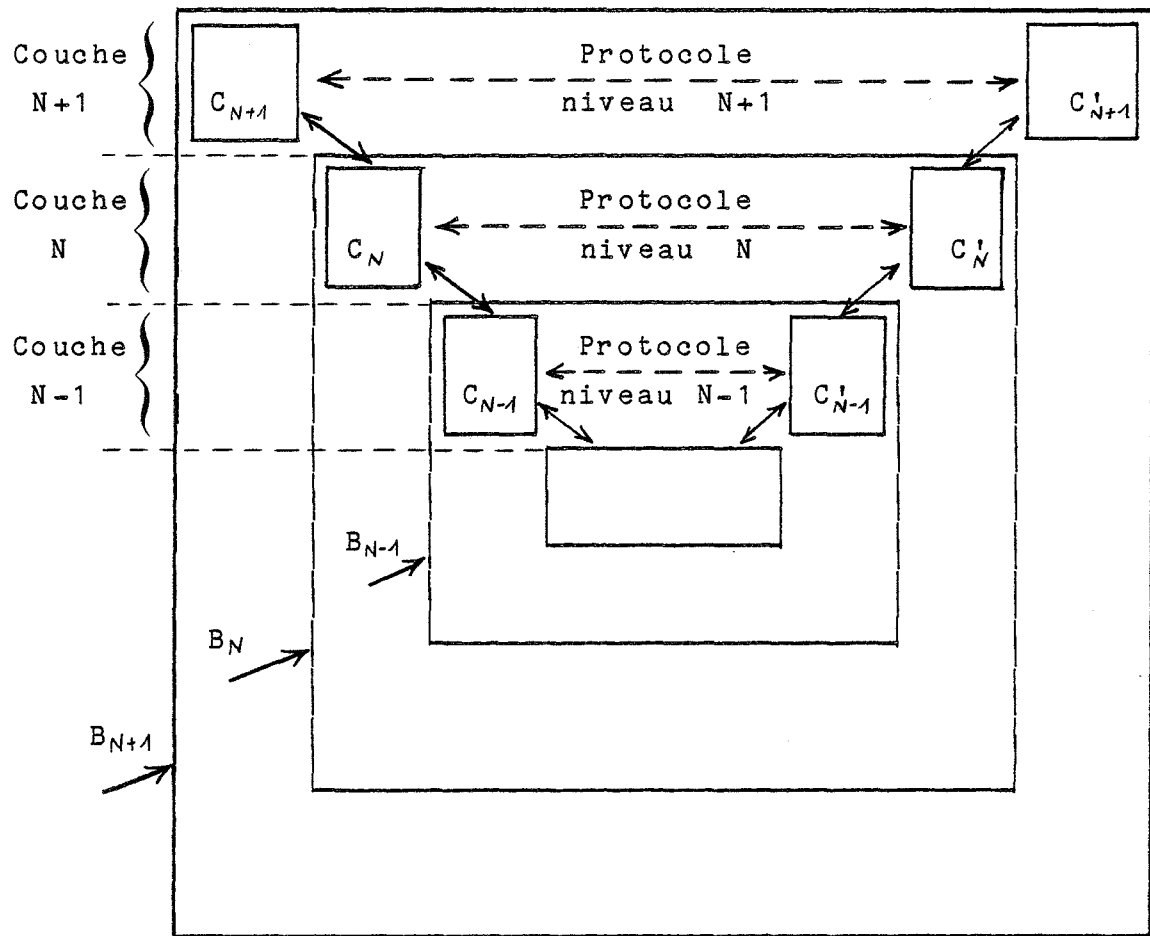
3. Le modèle de référence de l'ISO

3.1. Objectifs

Sans fournir une description précise des spécifications fonctionnelles des différents niveaux de protocoles d'interconnexion, le modèle de référence offre néanmoins un cadre de travail fonctionnel et conceptuel pouvant servir de bases à l'élaboration de normes.

Le modèle d'architecture proposé par l'ISO est fondé sur la structuration en couches. Celle-ci a l'avantage d'isoler les différents niveaux fonctionnels et de rendre les couches des niveaux supérieurs indépendantes des fluctuations des couches des niveaux inférieurs et inversement.

Chaque couche fonctionnelle définit une machine abstraite manipulable par un langage et capable d'exécuter une activité déterminée. La technique de structuration utilisée, technique d'emballage, consiste (ZIMMERMANN: RES-40) à considérer un ensemble d'entités comme une entité unique, sans chercher à connaître de l'extérieur la structure interne de cette entité composée qui n'est connue que par son comportement au travers de ses interfaces avec l'extérieur.



C_n composant adaptateur d'interface

Figure - 2. Technique de structuration

3.2. L'architecture à sept couches (ISO: MAS-12)

1) Couche physique

Elle assure l'interface avec le support physique et la commande des circuits de données.

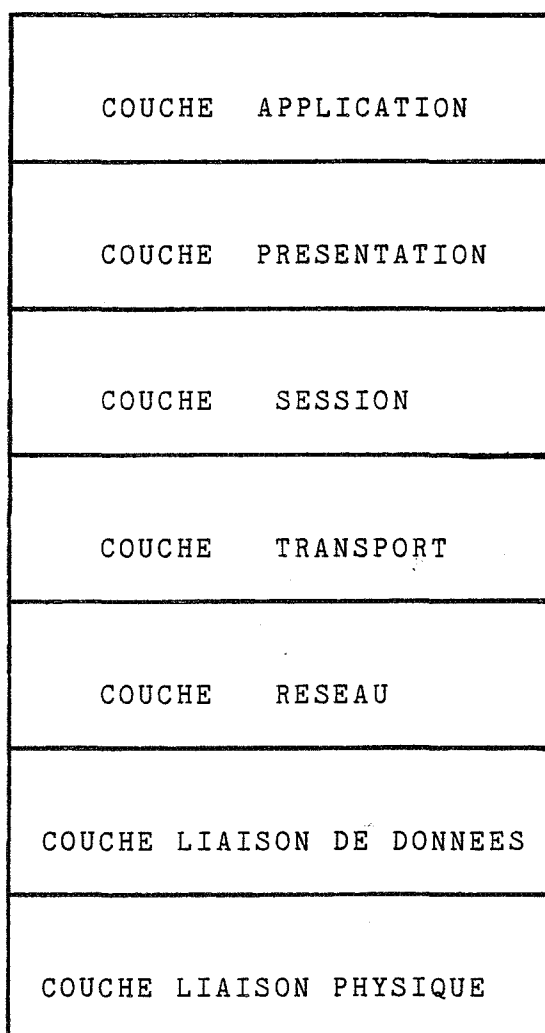


Figure - 3. Architecture à sept couches de l'ISO

2) Couche liaison de données

Elle assure la transmission sans erreurs de blocs de données en utilisant les liaisons physiques.

3) Couche réseau

Elle assure l'acheminement et le séquençement de paquets d'informations entre systèmes extrêmes.

4) Couche transport

Elle assure le contrôle du transport de l'information de bout en bout.

5) Couche session

Elle établit et contrôle le dialogue entre deux entités distantes.

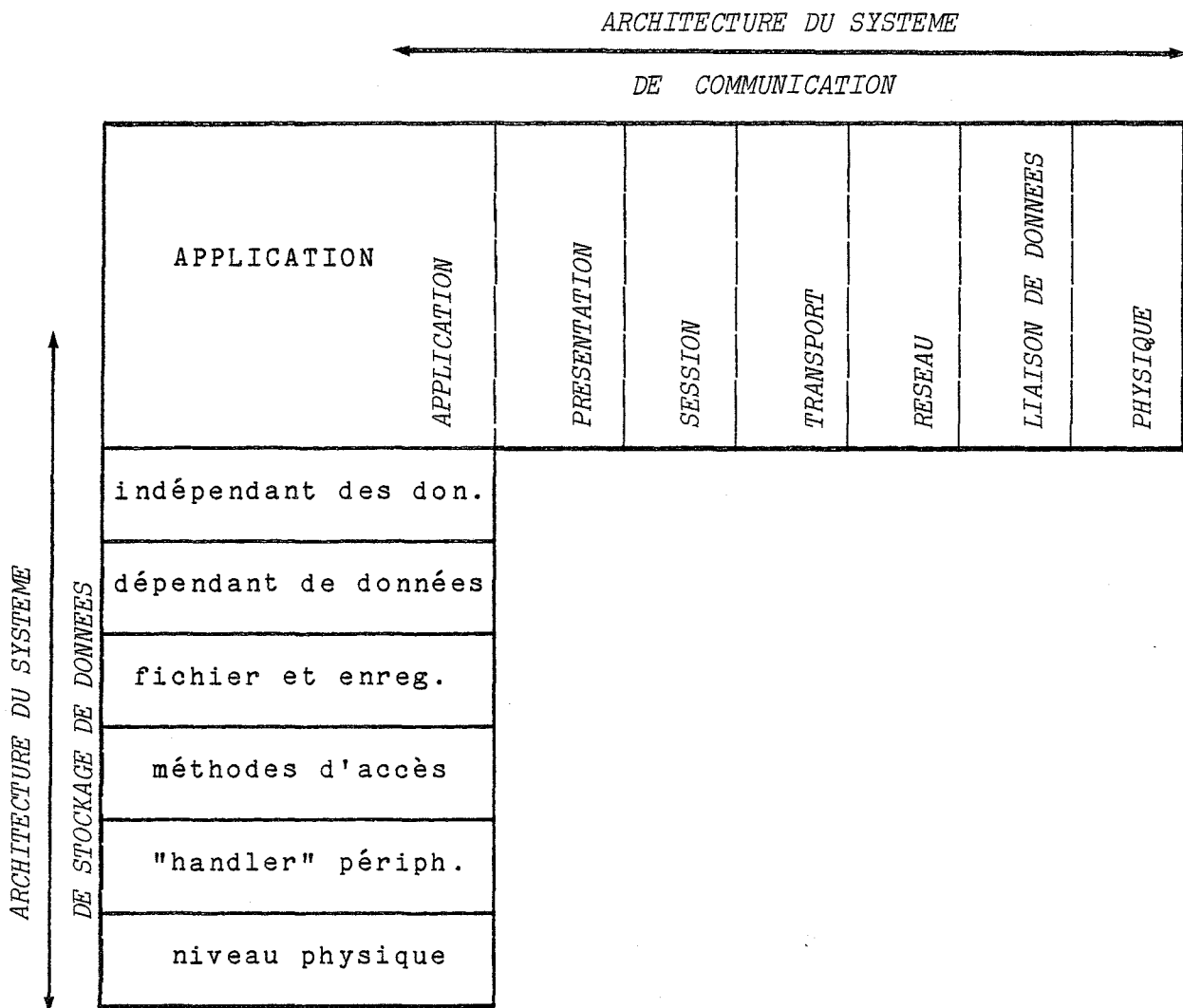
6) Couche présentation

Elle assure les mises au format adéquat des structures de données manipulées et leurs conversions éventuelles.

7) Couche application

Elle comprend les fonctions remplies par les applications qui coopèrent et les protocoles d'échange de niveau application.

4. Architecture orthogonale (BACHMAN: MAS-3)



Ce modèle d'architecture a l'avantage de combiner deux fonctions essentielles d'un système informatique en environnement réparti en les dissociant très nettement :

- le système de stockage de données,
- le système de communication.

5. Techniques de structuration

"La maîtrise des systèmes complexes que sont les réseaux informatiques passe par une décomposition des problèmes résultant d'une structuration judicieuse des systèmes répartis en sous-systèmes à leur tour décomposables en éléments plus simples jusqu'à parvenir à des concepts élémentaires directement réalisables." (ZIMMERMANN: RES-40)

Le type d'architecture le plus couramment rencontré est le résultat de l'application successive de la technique d'emballage pour former une architecture en couches.

5.1 Structuration par emballage ou par enveloppe

Considérons le service de la figure ci-après. Pour pouvoir accéder au service dans un environnement réseau, il est nécessaire de rajouter au service initial une couche de logiciel assurant l'interface avec le service.

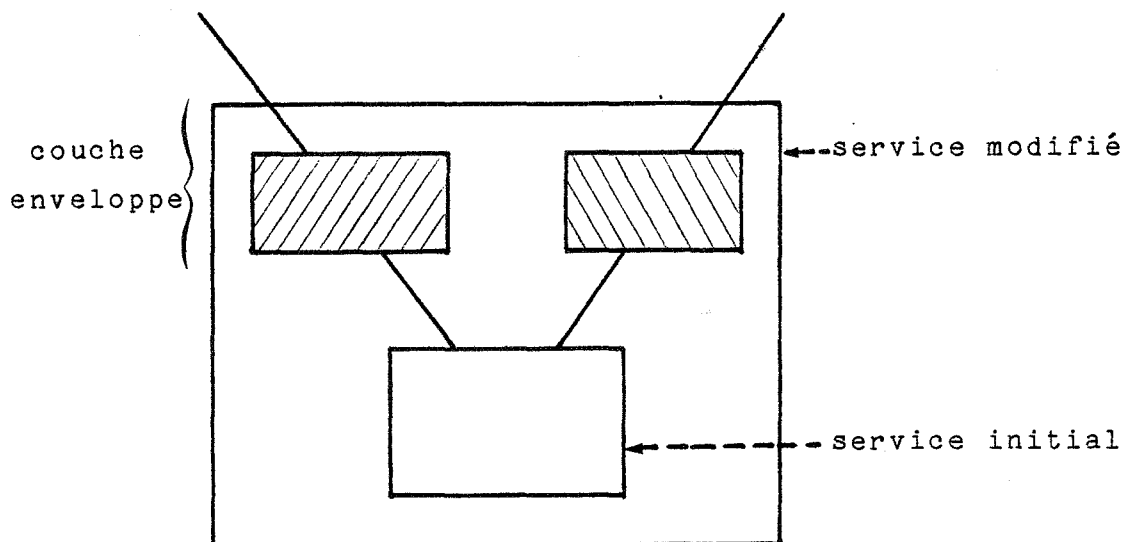


Figure - 4. Vue logique d'une enveloppe.

C'est cette couche de logiciel que nous appelons : enveloppe réseau du service. Cette couche peut être répartie pour permettre l'accès au service à partir de différents points du réseau.

Deux entités sont emballées par deux composants terminaux appartenant à un même niveau d'abstraction. Ces composants assurent la fonction grâce à un protocole de haut niveau.

Etant donné le caractère fermé des systèmes actuels, pour permettre, à des utilisateurs, l'accès distant de certains services, il s'avère souvent nécessaire de rajouter aux fonctions initiales une interface jouant le rôle d'enveloppe. Dans le cadre du niveau application, cette enveloppe a pour but essentiel d'être un adaptateur de commandes.

L'avantage de cette structuration réside dans le fait qu'on n'a pas besoin de modifier le système initial et l'architecture interne des services initiaux. Un exemple d'utilisation de cette technique de structuration est donné par l'application réseau (CHAMBON et al. RES-7) du Centre de Calcul Réparti.

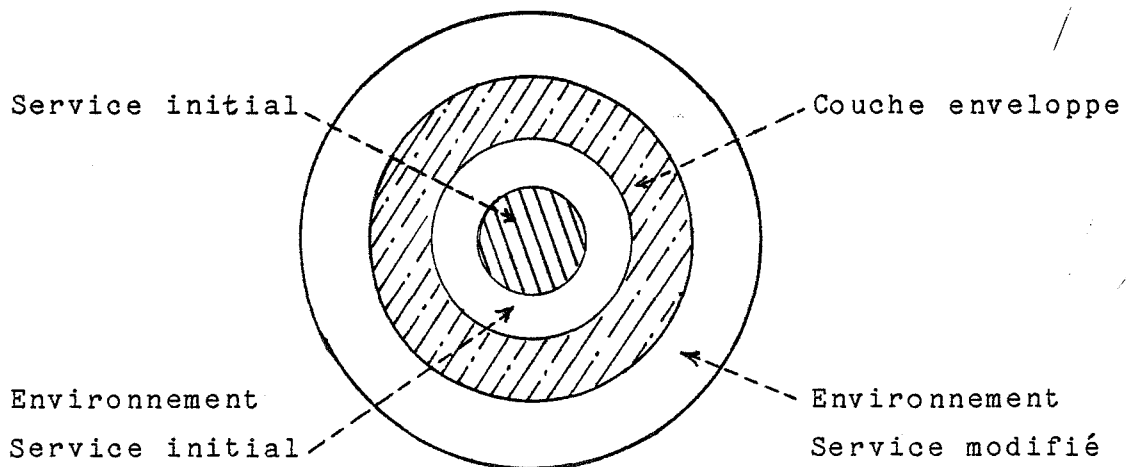


Figure - 5. Application Réseau du CCR.

5.2 Structuration en couches

5.2.1 Principe de la structuration en couches

C'est la technique la plus employée dans l'interconnexion des systèmes distribués. Elle permet de définir des niveaux relativement autonomes.

L'intérêt de rendre les couches indépendantes les unes des autres est considérable. La modification d'une couche n'influe pas sur les couches adjacentes à condition, bien entendu, que les services offerts par la nouvelle couche ainsi constituée et l'interface avec les couches voisines restent inchangés.

Dans les systèmes répartis, les entités du niveau le plus bas sont directement interconnectées pour former un réseau et coopèrent pour offrir un service de communication (GIEN, ZIMMERMANN: MAS-10) entre les différentes entités des niveaux plus élevés. Ces entités à leur tour coopèrent pour fournir des services à des entités de la couche au dessus. Elles le font en communiquant grâce aux services de communication fournis par la couche adjacente en dessous. Et le processus de récurrence se poursuit ainsi jusqu'à la couche du niveau le plus élevé où les entités coopèrent non pas pour fournir des services à des couches supérieures mais pour leur propre compte.

Néanmoins dans la structuration en couches, les couches internes n'ont pas pour seul objectif de servir les couches les plus externes mais doivent constituer de véritables machines abstraites ayant un sens et utilisables par d'autres systèmes.

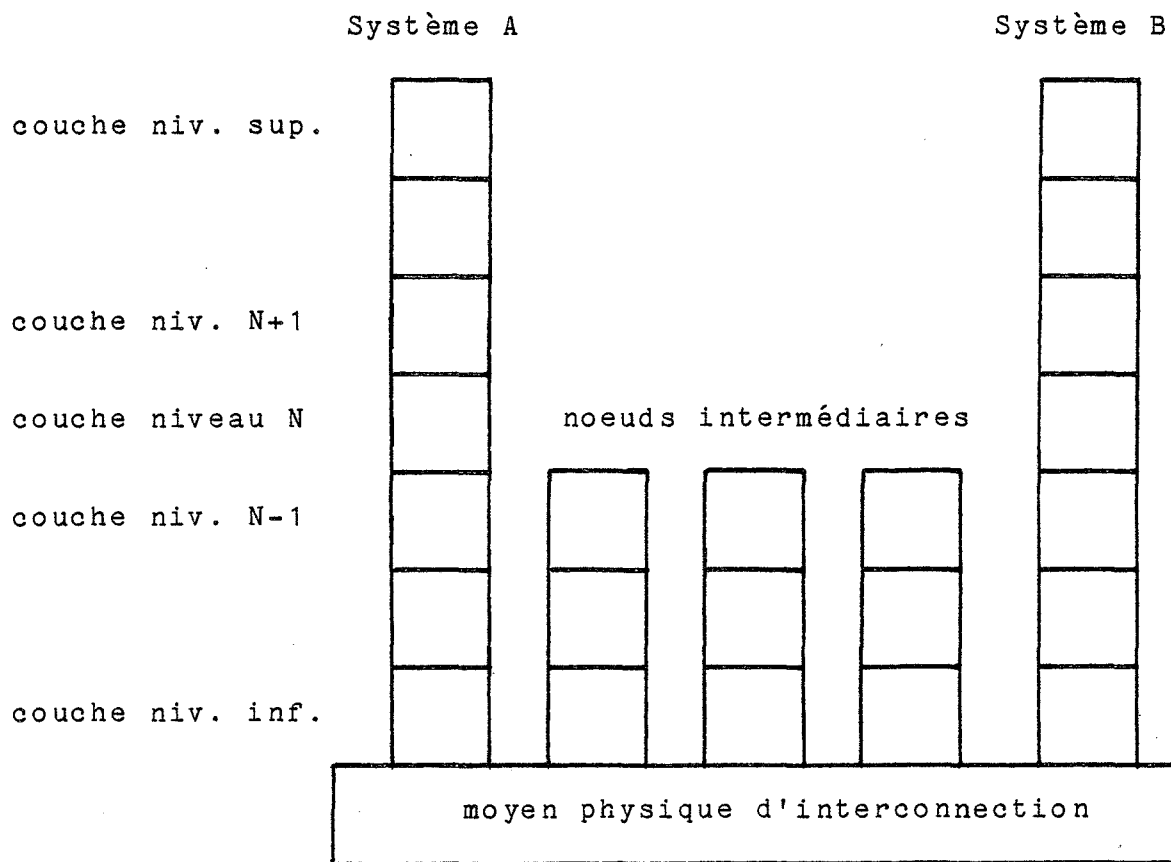


Figure - 6. Exemple de structuration en Couches.

En fait la structuration en couches peut être vue comme une forme récurrente de la structuration par emballage.

Il est important de souligner aussi que dans cette technique, un objet n'est connu à l'extérieur que grâce à son langage de manipulation. En d'autres termes la structure interne de l'objet n'est pas connue à l'extérieur de cet objet.

5.2.2 Problème de communication

Le but de la structuration en couches est, à partir d'un certain niveau fonctionnel (niveau d'organisation), de former d'autres niveaux fonctionnels plus complexes que les précédents. Dès lors se pose un problème de communication (mise en commun de l'information) entre les différents niveaux.

Mais pour mieux appréhender ces problèmes, nous distinguerons :

- la communication entre éléments de même niveau fonctionnel régie par des protocoles de bout en bout quoiqu'il puisse être nécessaire de faire des interconnexions en cascades;
- la communication entre éléments de niveaux d'organisation différents. Cette communication est le plus souvent réalisée par le moyen d'interfaces offrant un langage de manipulation des objets de niveaux inférieurs (ou plus généralement des machines de niveaux inférieurs) par les objets de niveaux supérieurs.

Plus le protocole est de haut niveau, plus le langage est dense au plan sémantique.

6. Quelques approches de la répartition des données

Dans la conception des systèmes répartis, on peut distinguer deux approches:

- une approche "séparatrice"
- une approche "unificatrice"

6.1. L'approche séparatrice

Elle consiste à faire à priori une différentiation entre application centralisée et application répartie. Dès lors, elle fournit des primitives spéciales d'accès à distance qui sont pour la plupart différentes des primitives d'accès local.

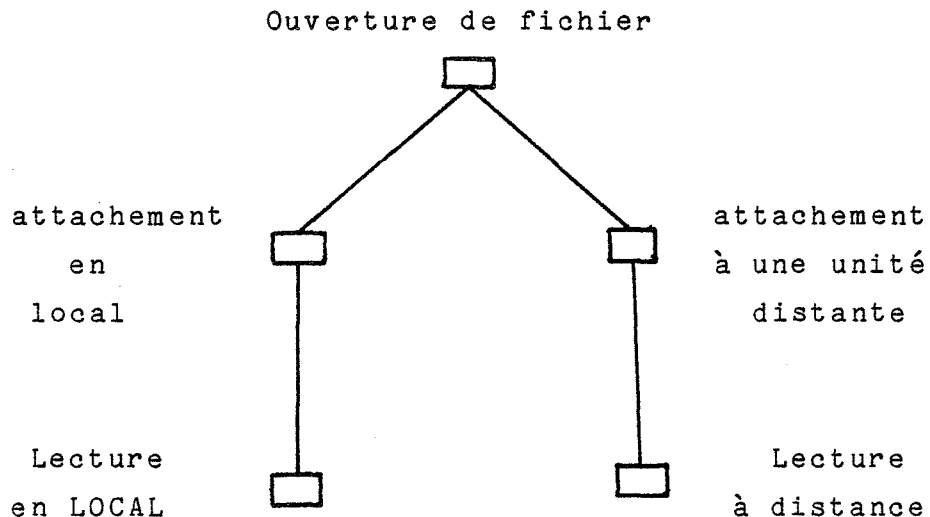
Dans cette approche la liaison est faite, au départ, au moment de l'assignation. Elle est statique. Les primitives d'accès émises, le sont dans un environnement bien déterminé, même si ce dernier n'est pas l'environnement local. On perd donc tout le dynamisme qui existait au niveau d'une liaison différée.

De plus, les utilisateurs sont obligés de connaître deux (voire plusieurs, si les systèmes en jeu ne sont pas homogènes) types de primitives d'accès aux données.

6.2. L'approche unificatrice

Elle a été faite dans les bases de données réparties (GARDARIN: MAS-7) en isolant un noyau de communication. L'application de ce principe aux systèmes répartis consiste à ne pas distinguer à priori une application répartie des autres applications qui ne le sont pas.

Cette approche ramène la répartition à des fonctions de topographie.



6.3. Notre approche : un palliatif

Elle n'est exactement ni la première ni la seconde approche ; mais un palliatif qui permet de faire croire à l'utilisateur qu'il se trouve dans la deuxième approche.

Elle consiste à anticiper sur le moment de liaison et à créer toutes les conditions pour qu'au moment de cette liaison tout se passe comme si toutes les ressources nécessaires étaient locales.

Pour réaliser l'interface et les mécanismes, dans le cadre de cette approche, nous serons amenés à mieux expliciter les couches session, présentation et application

du modèle de référence et à en donner une architecture applicable à notre système.

Cependant, l'interface ainsi constituée est par essence provisoire. Elle disparaîtra dès l'instant où les systèmes seront conçus de manière à être ouverts à des environnements extérieurs aux leurs.

En effet, il est tout à fait concevable de penser que les systèmes d'exploitation futurs rempliront eux-mêmes les fonctions actuellement confiées à l'interface "système de coopération". Le système d'exploitation standard pourra, à partir de la référence à un fichier, d'une part consulter son environnement et d'autre part s'adresser, le cas échéant, à d'autres systèmes pour obtenir l'accès à ce fichier distant comme s'il s'agissait de son propre environnement. Des réalisations allant dans ce sens sont en cours dans les réseaux DECNET (Phase III). Mais l'inconvénient majeur en est la dépendance vis à vis d'un seul constructeur.

Il est souhaitable de pouvoir disposer de réseaux hétérogènes et de possibilités de choix entre matériels provenant de constructeurs différents. Ceci sera rendu possible dans l'hypothèse d'une normalisation d'un certain nombre de couches du modèle ISO. Dès lors l'opportunité d'un système de coopération comme le nôtre disparaît.

CHAPITRE V

MECANISMES D'EXECUTION REPARTIE

1. Communication entre applications
2. Langages de communication
 - 2.1. Différentes solutions au problème de traduction
 - 2.2. Critères de choix d'un langage de commande
 - 2.3. Hiérarchie de niveaux
3. Techniques réseaux
4. Protocole session
 - 4.1. Services offerts
 - 4.2. Principe de la communication interprocessus
5. Protocole présentation
 - 5.1. But de la couche présentation
 - 5.2. Les protocoles offerts

Les techniques de structuration et les modèles d'architecture de systèmes permettent surtout de définir les différents composants structurels des systèmes répartis (couches, interfaces, modules).

Pour étudier les aspects dynamiques de ces systèmes, il nous faut recourir à la notion d'activité en donnant vie (GUIBOUD-RIBAUD: MAS-11) aux objets composant un niveau.

Une activité répartie est considérée comme étant logiquement composée d'éléments distincts appelés entités et pouvant communiquer ou plus généralement interagir les uns avec les autres au moyen d'une interface (ZIMMERMANN: RES-40).

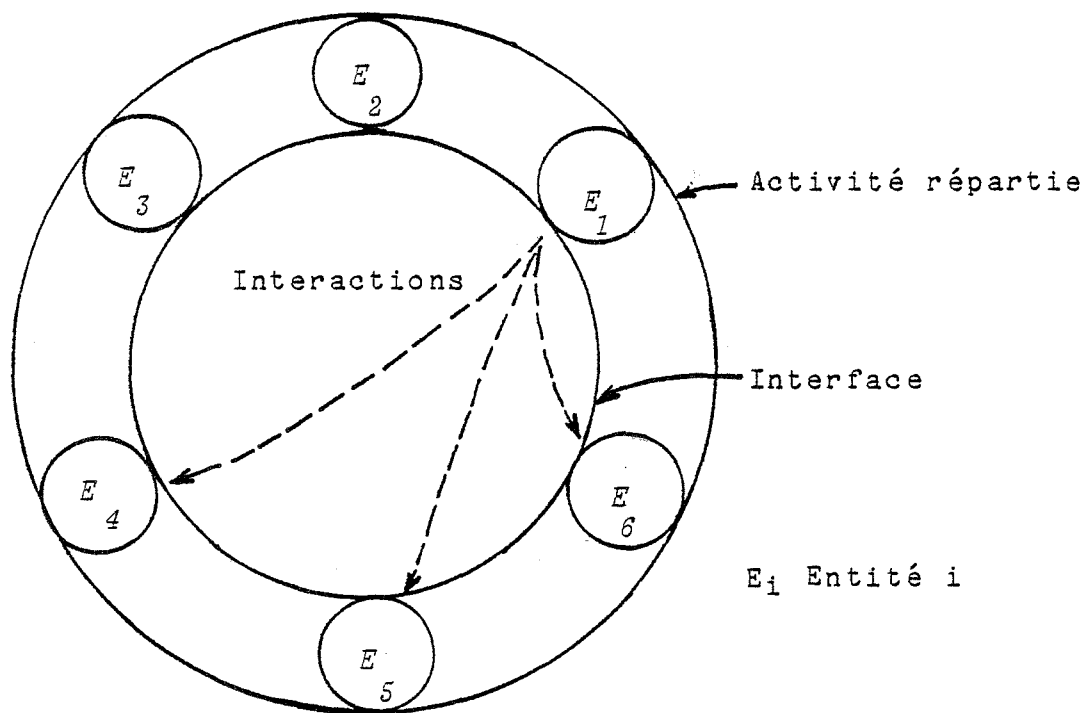


Figure - 1. Activité répartie

L'interaction ou la coopération entre entités ne peut se concevoir que par échange d'information. Cette opération implique donc la communication (LE LANN: CAC-9) qui s'effectue physiquement par l'intermédiaire d'objets appelés couramment messages .

Pour que les entités d'une activité répartie coopèrent harmonieusement au déroulement de l'activité commune, il faut qu'elles suivent dans leurs interactions un ensemble de règles qu'on appelle leur PROTOCOLE.

On dira donc qu'une activité est réalisée par un ensemble d'entités qui interagissent au moyen d'une interface suivant un protocole.

Un système réparti étant caractérisé par l'absence d'une connaissance instantanée de l'état global du système, on se heurte au niveau de la communication à ce problème crucial.

1. Communication entre applications

La communication constitue, pour une entité, l'unique moyen d'agir ailleurs. Ailleurs dans l'espace, c'est-à-dire sur une autre entité. Ailleurs dans le temps, soit sur une autre entité, soit localement (LE LANN: CAC-9).

Dans les systèmes répartis plus que dans les systèmes centralisés, le problème de la communication se pose avec beaucoup d'acuité. En effet, on conçoit bien que dans les systèmes centralisés, la communication puisse s'effectuer par partage de données communes voire par partage d'espace mémoire commun. Par contre, on conçoit mal une simple transposition dans les systèmes répartis car

l'espace d'adressage est distinct d'une machine à l'autre et il n'y a pas de zone mémoire commune. Il est donc impossible de recourir à ce moyen pour faire communiquer deux applications qui coopèrent.

Un des autres problèmes de la communication dans les systèmes répartis est la dualité qui existe entre :

- la communication entre applications s'exécutant sur le même site ; celle-ci est le plus souvent réalisée au moyen d'appels de procédures avec passages de paramètres ;
- la communication entre applications (ou parties d'une même application) s'exécutant sur au moins deux sites différents. Cette dernière utilise nécessairement le service de transport des architectures actuelles.

Un moyen d'arriver à une certaine uniformisation dans les mécanismes d'échanges de données consiste à contraindre toutes les entités d'un système désirant communiquer à s'échanger explicitement des messages (CART: RES-5).

Cette solution, valable aussi bien dans les systèmes répartis que dans les systèmes centralisés, leur confère une grande extensibilité.

La notion de message est en effet essentielle. Elle a permis de résoudre de nombreux problèmes de communication dans les réseaux d'ordinateurs et dans les dialogues interprocessus des systèmes répartis.

Apparue dans les années 1960, la commutation de messages a donné naissance à la commutation par paquets (BARAN: RES-2). Celle-ci a été expérimentée à partir de 1969 sur le réseau ARPANET puis sur le réseau CYCLADES (POUZIN: RES-31).

Aujourd'hui des réseaux publics comme TRANSPAC (PONCET, ALLAIRE: RES-29) utilisent la commutation par paquets. Mais ce n'est qu'un moyen de réaliser des communications de messages.

Dans les systèmes répartis, le service de transport apparaît comme un commutateur de MESSAGES (CART: RES-5). Sa fonction est en effet de recueillir les messages des émetteurs et de les véhiculer jusqu'aux destinataires. Quant aux entités qui dialoguent, elles synchronisent leurs actions par le biais des messages échangés : un message étant à la fois un véhicule d'information et un moyen de synchronisation.

2. Langage de requêtes

2.1. Différentes solutions au problème de traduction

Dans la communication entre systèmes hétérogènes, on est souvent amené à faire le choix d'un protocole de communication pour exprimer et véhiculer les requêtes en fonction des contraintes des systèmes participants et de la nécessité d'une ouverture pour assurer une certaine évolutivité n'entraînant pas la remise en cause de l'ensemble.

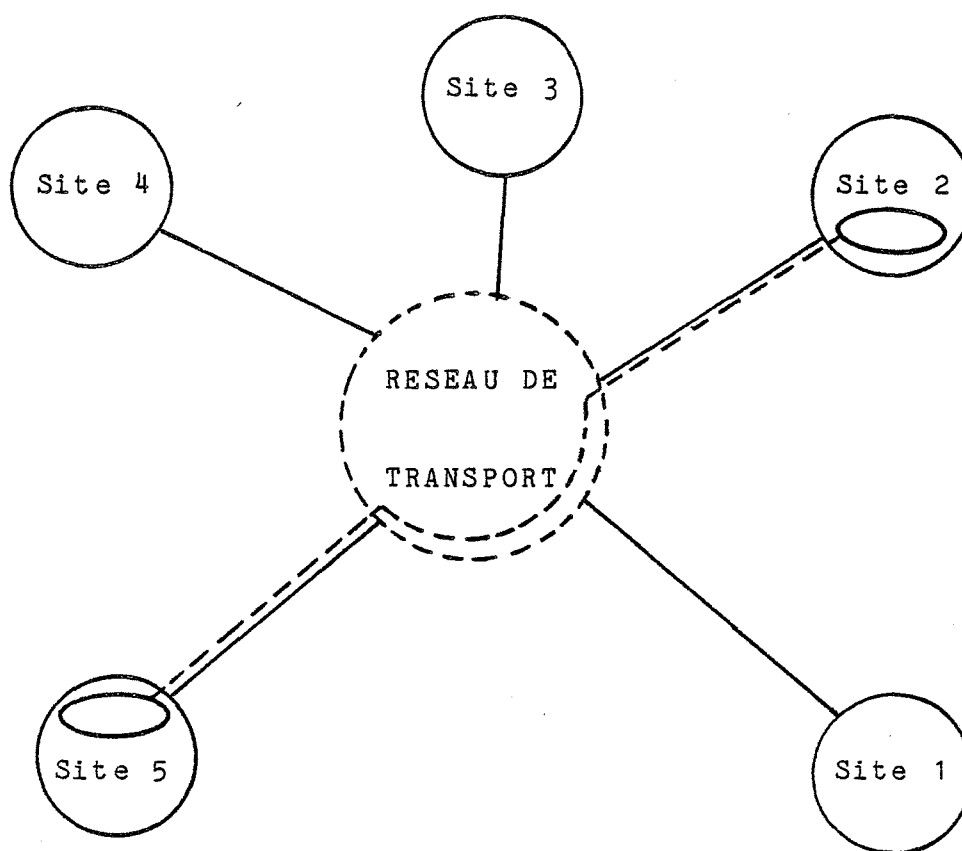


Figure - 2.

Concrètement :

- Faut-il à chaque interrogation d'un site lui envoyer toutes les requêtes dans un langage qui lui est propre ?
- Ou au contraire vaut-il mieux opter pour un langage unique suffisamment puissant pour exprimer toutes les requêtes des différents sites particuliers ?

a) Plusieurs langages

* Contrôle réparti :

Le processus qui utilise le système réparti (processus de soumission de requêtes implanté sur chaque site) s'adresse à chacune des entités réparties dans le langage propre au site interrogé. Cette solution oblige à implanter sur chaque site un interprète ou un traducteur fournissant tous les langages des systèmes participants.

Quand on rajoute un système, il faut modifier sur chaque site le processus de soumission de requêtes pour qu'il connaisse le nouveau venu.

* Contrôle centralisé :

Dans un système à contrôle centralisé, il faut mettre en place un traducteur universel (dans la limite des langages propres aux sites du réseau) sur le site maître qui envoie à chacun des satellites un message dans son langage de commande particulier.

Cette démarche accroît la charge du maître et rend le système global très vulnérable car le site maître devient un goulot d'étranglement. Dès qu'il tombe en panne l'application répartie devient inutilisable.

b) Langage unique

Le processus de soumission de requêtes s'adresse à tous les serveurs dans un langage unique qui est celui du système de coopération.

Il incombe à chaque serveur la tâche de traduire les requêtes du langage général en commandes spécifiques du système considéré.

Il n'y a plus qu' un seul langage de commande qui est véhiculé par le réseau de transport. Chaque serveur du système de coopération n'a plus qu'à connaître les caractéristiques du système sur le site où il est implanté.

Il ne fait pas de doute qu'il est préférable, sauf cas très particulier, d'adopter cette seconde solution à cause des avantages qu'elle offre par rapport à la première aussi bien du point de vue de la répartition de la charge que de la résistance aux pannes.

Une illustration de cette démarche est donnée par le système SIGOR (DANG et al. MER-4) qui est le prolongement de l'Interprète Généralisé Orienté Réseau IGOR (SERGEANT, FARZA: MER-15). Le système SIGOR, construit sur le réseau CYCLADES, propose un modèle de machine réseau logique ayant un langage de commande unique pour l'expression des applications réparties.

La machine réseau SIGOR se compose sur chaque site participant d'un sous-système interpréteur et de mécanismes d'aide à la répartition.

2.2. Critères de choix d'un langage de commande

On s'intéresse ici au contenu de la communication c'est-à-dire à la requête véhiculée. A ce stade le problème qui se pose est d'abord de savoir s'il faut choisir un langage assez puissant et assez général pour pouvoir convenir à toute une famille d'applications réparties ou alors adopter un langage spécifique au système de coopération. Pour ce faire nous disposons de premiers critères de choix :

a) Famille de langages

* avantages :

- homogénéité des langages utilisés dans le système réparti;
- réutilisation des traducteurs ou d'une partie des traducteurs développés pour d'autres applications;

* inconvénients :

- lourdeur et volume des traducteurs;
- faibles performances.

b) Langage spécifique

* avantages :

- langage mieux adapté aux traitements de fichiers,
- taille du traducteur plus modeste ;

* inconvénients :

- langage strictement orienté fichier et non utilisable par d'autres applications ;
- obligation de développer entièrement tous les traducteurs nécessaires.

A partir de cela nous pouvons opter pour une des solutions avec sans doute un compromis. Il nous reste ensuite, pour être complet, à déterminer la syntaxe et la représentation de ce langage (paramètres positionnels ou à mots clés, codage des différents champs).

2.3. Hiérarchie de niveaux

Sans nous préoccuper pour l'instant du système de stockage de l'information, nous allons examiner l'architecture du système de communication.

Supposons le cas de deux applications situées sur deux sites différents qui coopèrent pour échanger un fichier. L'application située sur le site émetteur envoie le fichier article par article.

On voit, d'après le schéma suivant, que de proche en proche, chaque couche du site émetteur ajoute des éléments d'informations au message initial constitué par l'article du

e_i : éléments de protocole

P_i : Protocole de niveau i

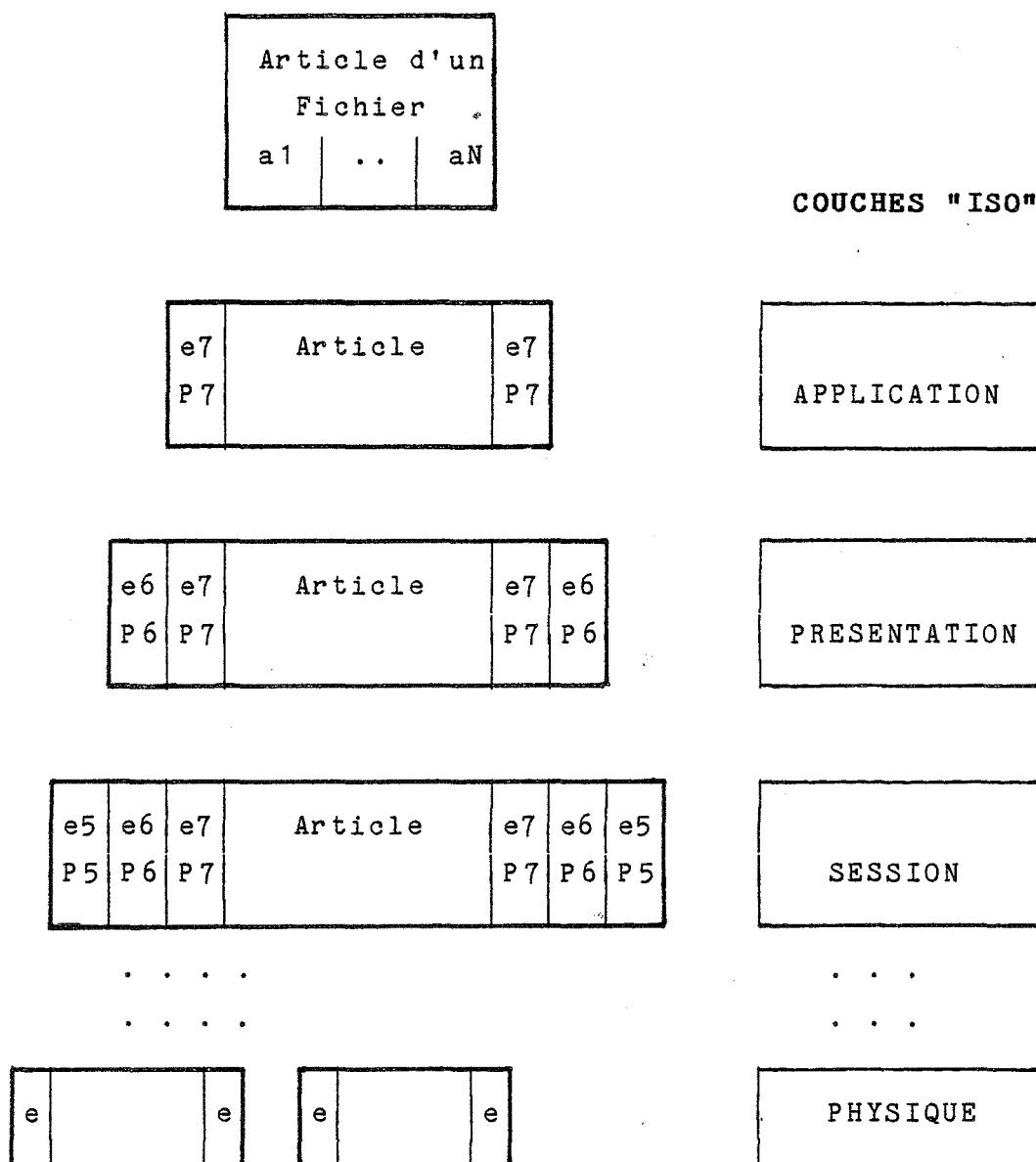


Figure - 3. Hiérarchie de niveaux de protocoles

fichier. Ces informations constituent en fait des éléments de protocoles qui, selon le niveau considéré, peuvent servir à indiquer :

- l'adresse d'un destinataire (processeur physique ou logiciel),
- la taille d'un message ou d'un fragment de message,
- la signification de certains types de données.

Ces différents éléments d'informations ne sont significatifs que pour les deux entités correspondantes du protocole considéré.

Remarquons toute la quantité d'informations qu'il faut adjoindre à l'article du fichier initial sans compter les informations de contrôle et de supervision des différentes voies de communication (comme les négociations pour établir ou rompre une liaison par exemple).

3. Techniques réseaux

Nous regroupons sous cette rubrique les quatre niveaux:

- couche transport,
- couche réseau,
- couche liaison de données,
- couche liaison physique

telles qu'elles sont définies dans l'architecture ISO. Rappelons que ces couches ont pour but essentiel d'assurer le routage et le contrôle du transport de l'information de bout en bout.

L'un des objectifs du découpage fonctionnel en couches étant de rendre celles-ci indépendantes les unes des autres, nous n'insisterons pas sur les protocoles de bas niveau. Nous nous contenterons d'énumérer les services nécessaires au niveau de la couche session. Ces services sont fournis par la couche transport et les couches inférieures à celles-ci.

4. Protocole de niveau session

Le but de la couche session est d'aider à supporter les interactions entre deux entités de niveau application qui coopèrent.

4.1. Services offerts

La couche session offre deux types de service :

* établissement et rupture de la liaison entre deux entités de niveau présentation,

- établissement de la liaison,
- utilisation (dialogue plus ou moins long)
- rupture de la liaison;

* opérations de contrôle, synchronisation et de séquençement des données échangées.

4.2. Principe de la communication interprocessus

Pour assurer le transfert de données entre deux entités de niveau présentation, la couche session offre un service de communication interprocessus. Les processus distants communiquent entre eux au moyen de connexions unidirectionnelles et virtuelles : les voies logiques. La voie logique écoule des messages entre deux processus de telle sorte que les messages soient bien reçus dans l'ordre de leur émission. La voie logique permet aussi le contrôle du flux des messages.

5. Niveau présentation

5.1. But de la couche présentation

Cette couche doit permettre une description des structures de fichiers autorisant des conversions en fournissant un langage de description de structures. A chaque type de fichier doit correspondre une description dans le langage de description de structures.

5.2. Les protocoles offerts

Ils opèrent au niveau de la couche présentation du modèle de référence de l'ISO. Ils sont en partie intéressés par le contenu du message. Ces protocoles sont assez souvent spécifiques de l'application envisagée.

Les protocoles suivants ont été définis sur le réseau ARPA (LOUIT: MER-12) :

- protocole de transfert de données spécifiant les méthodes de formatage de données,
- protocole de transfert de fichiers,
- protocole de transfert d'informations de visualisation,
- protocole de boîte à lettres.

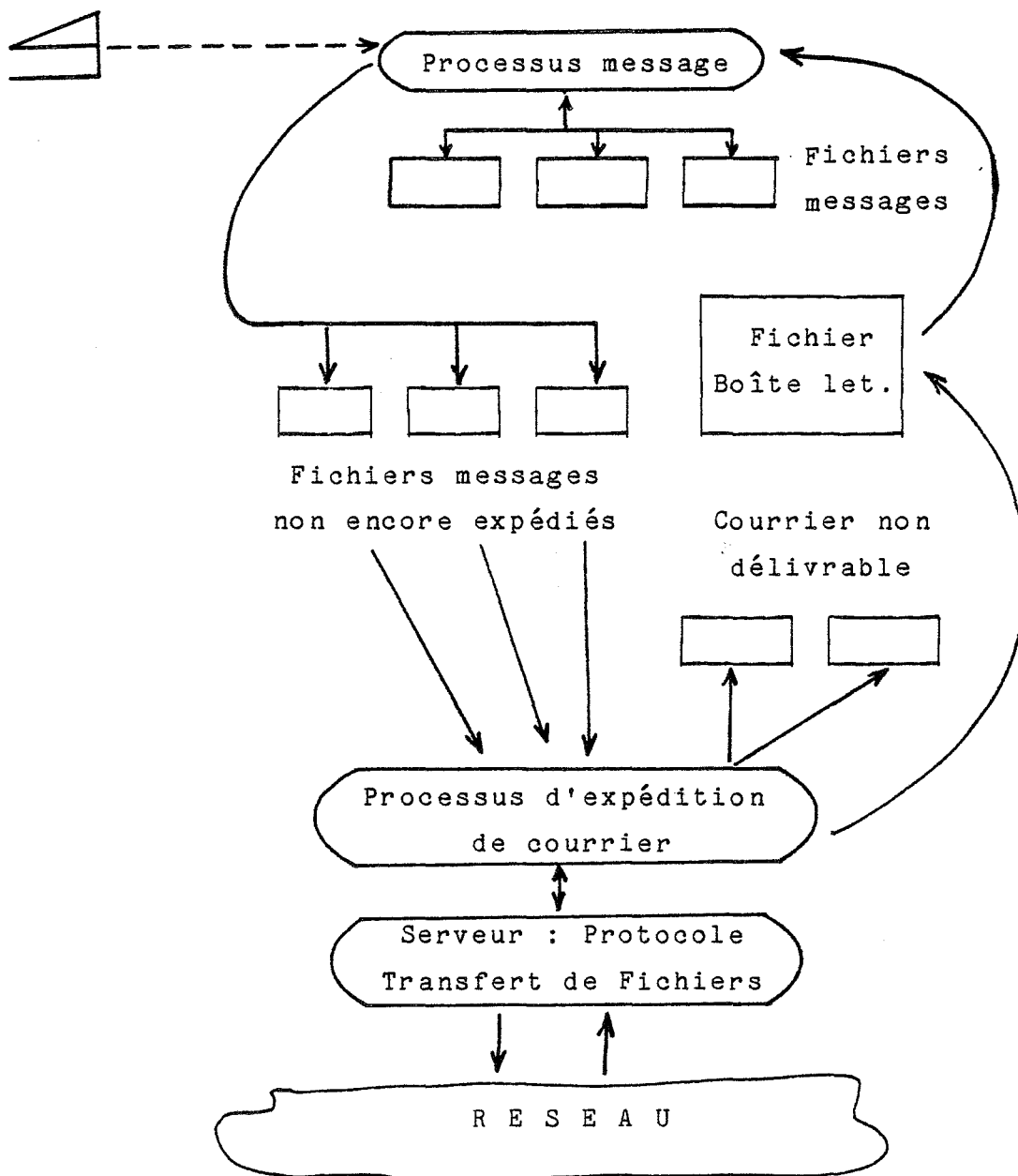


Figure - 4. Système de messagerie sur ARPA

CHAPITRE VI

LE SYSTEME DE COOPERATION DE SGF

1. Intérêt et objectif d'un SCSGF
 - 1.1. Augmentation potentielles des ressources
 - 1.2. Coopération de systèmes hétérogènes
 - 1.3. Spécificité par rapport aux systèmes de fichiers distribués existants
2. Modèle de coopération choisi
 - 2.1. Composants du SCSGF
 - 2.2. Types de dialogues possibles
 - 2.3. Gestion de catalogue
3. Mécanisme de fonctionnement du Système de Coopération
 - 3.1. Localisation des fichiers
 - 3.2. Espace de désignation
 - 3.3. Contrôle de concurrence d'accès
4. Le Système de Coopération proposé
 - 4.1. Caractéristique
 - 4.2. Méthodologie
 - 4.3. Apport du SCSGF

1. Intérêt et objectif

La finalité du Système de Coopération de Systèmes de Gestion de Fichiers (SCSGF) est d'assurer la transparence de la localisation des ressources fichiers en garantissant le maximum d'autonomie au SGF local à chaque site (JUREMA, NIANG: RES-22). Conçu dans le cadre du Centre de Calcul Réparti (Chapitre VIII), le SCSGF a pour buts principaux:

- d'accroître les ressources potentielles de chaque site, à travers une gestion globale de l'espace de stockage du réseau local;
- de faire coopérer des systèmes hétérogènes en respectant les caractéristiques et l'indépendance de chacun.

1.1. Augmentation des ressources potentielles

C'est là un objectif important de l'application. Pour nous en effet l'intérêt primordial de la coopération réside dans le fait qu'un site puisse mettre ses ressources à la disposition d'un autre et pallier ainsi une éventuelle insuffisance locale. Cette aide est entièrement transparente à l'utilisateur qui dès lors n'a plus à se soucier du lieu de localisation de son fichier.

Un site donné n'aura recours aux ressources d'un autre site que lorsqu'il ne pourra plus satisfaire la demande en local. Dès lors un dialogue s'établit pour allouer des ressources distantes. Mais en définitive, c'est

au SGF (Système de Gestion de Fichiers) distant que revient la tâche de superviser l'utilisation de ses propres ressources.

1.2. Coopération de systèmes hétérogènes

Notre but n'est pas de construire de toutes pièces un SGF mais plutôt de profiter de ceux qui existent pour enrichir potentiellement chaque site. Chaque SGF a sa propre façon de structurer les informations et de les stocker. Il est le seul maître de la manipulation et de l'allocation des ses ressources. D'où un grand effort pour maintenir les caractéristiques et l'indépendance de chaque site.

Cet effort s'avère très profitable dans la mesure où la présence ou l'absence d'un site dans le réseau doit se traduire par une disponibilité plus ou moins grande de ressources potentielles (CHAMBON et al.: RES-8, RES-9). Dans le cas du CCR (Centre de Calcul Réparti), par exemple, l'exclusion d'un ordinateur participant n'affecte pas le fonctionnement général du système. Seuls les utilisateurs directement liés à cette machine s'apercevront de cette défaillance. Au niveau des entités qui coopèrent, l'absence d'un site donné ne représente que l'absence d'une aide potentielle du point de vue des ressources qui pourraient être allouées aux autres sites.

1.3. Spécificité par rapport aux systèmes de fichiers distribués existants

Pour profiter pleinement des avantages de chaque SGF, il nous faut promouvoir une véritable coopération entre eux. Or, la plupart des systèmes existant en ce domaine ont

une approche différente du problème. Le produit MADRE (Chapitre II §3), vu plus haut, est en fait, une méthode d'accès réseau qui offre à l'utilisateur les moyens d'accéder à un fichier distant (CHUPIN, SEGUIN: GFR-2). Quant au DAP (Chapitre II §4), il est conçu pour l'architecture distribuée DNA (Digital Network Architecture) et s'intègre dans un environnement homogène constitué de PDP 11. En plus, d'une méthode d'accès réparti, il offre des utilitaires de transferts de fichiers (DECNET: GFR-3).

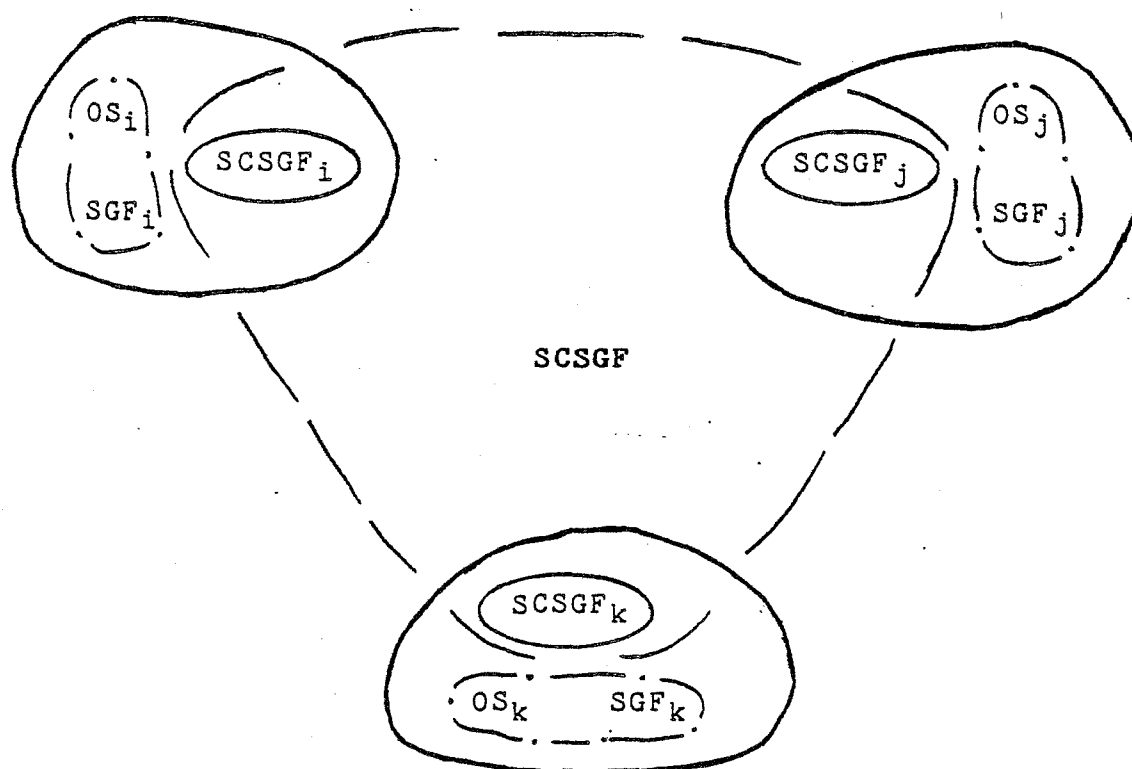
Quant à nous, notre Système de Coopération a un environnement hétérogène et veut bénéficier des avantages de chaque SGF constructeur. Mais l'originalité de cette application réside dans le fait qu'elle s'insère dans le projet d'ensemble du CCR dont elle bénéficie de tous les avantages, aussi bien matériels que logiciels. En effet, les terminaux sont banalisés et raccordés logiquement à un ensemble de "fonctions frontales" mettant en relation des utilisateurs avec des services localisés sur différentes machines. Ces utilisateurs n'ont à connaître qu'un langage de commande unique pour tout le CCR: le Langage de Commande Externe (LCE). Ces acquis du CCR ont donc induit dans une large mesure le choix de notre modèle de coopération.

2. Modèle de coopération choisi

Nous avons été conduits à opter pour une méthode de coopération où les sites restent autonomes et ne sont pas privilégiés les uns par rapport aux autres. Le logiciel chargé de promouvoir cette coopération (logiciel que nous appellerons désormais Système de Coopération de SGF: SCSGF) est réparti sur les différents sites participants du réseau.

2.1. Composants du SCSGF

Le Système de Coopération de SGF (SCSGF) est une entité composée qui réalise une activité répartie. Elle se décompose logiquement en différentes sous-entités qui communiquent entre elles et échangent de l'information. Ceci est réalisé grâce à un ensemble de règles et de conventions qui sont des protocoles propres au système de coopération.



SGSGF: Système de coopération de SGF

SGF_i : Système de Gestion de Fichiers standard
du site i

OS_i : Système d'exploitation du site i

Figure - 1. Architecture du SCSGF.

Le SCSGF sert d'interface entre les différents SGF. Il est situé à un niveau où il reçoit des commandes: celles-ci sont globales et se rapportent au fichier dans son

unité; d'où l'approche transfert de fichiers que nous avons choisie. Pris comme une entité logique globale, le système de coopération peut être considéré comme une machine abstraite d'accès à des fichiers locaux et distants.

Dans un cas comme le nôtre, l'approche transfert de fichiers est amplement suffisante, dans la mesure où très souvent le fichier se trouve dans l'environnement du service qui l'a créé ou du service qui doit l'utiliser.

Si l'on considère les SGF locaux des constructeurs seuls, il est difficile de penser qu'ils puissent coopérer d'une façon directe dans la mesure où ils s'ignorent totalement. Les requêtes du SGF local (READ, WRITE, ...) ne seront nullement véhiculées à travers le réseau. Ces requêtes sont, quoi qu'il arrive, traitées en local sur le site en question. Néanmoins, si l'on considère l'ensemble constitué sur un site par:

- le SGF local du constructeur plus les fichiers gérés par lui sur ce site,
- le sous-système fonctionnel $SCSGF_i$ (Sous-Système de Coopération de SGF implanté sur le site i),

on peut parler dans ce cas-là de coopération, le composant fonctionnel $SGSGF_i$ constituant en quelque sorte une enveloppe réseau au SGF local. La macro application ainsi formée (SGF plus enveloppe réseau) dialogue bien avec ses homologues des autres sites suivant un protocole et des conventions bien définies.

Loin de s'ignorer, les divers sous-systèmes (ou sous-entités) sont, en fait, différents composants d'un Système de Coopération: le SCSGF. Le SCSGF est la vision

globale qu'on a de cette application. Vu par l'utilisateur, l'application SCSGF donne l'image d'un système offrant beaucoup plus de possibilités et de ressources que tout autre SGF pris individuellement. Il faut souligner que, là aussi, l'objet d'un dialogue c'est le fichier entier:

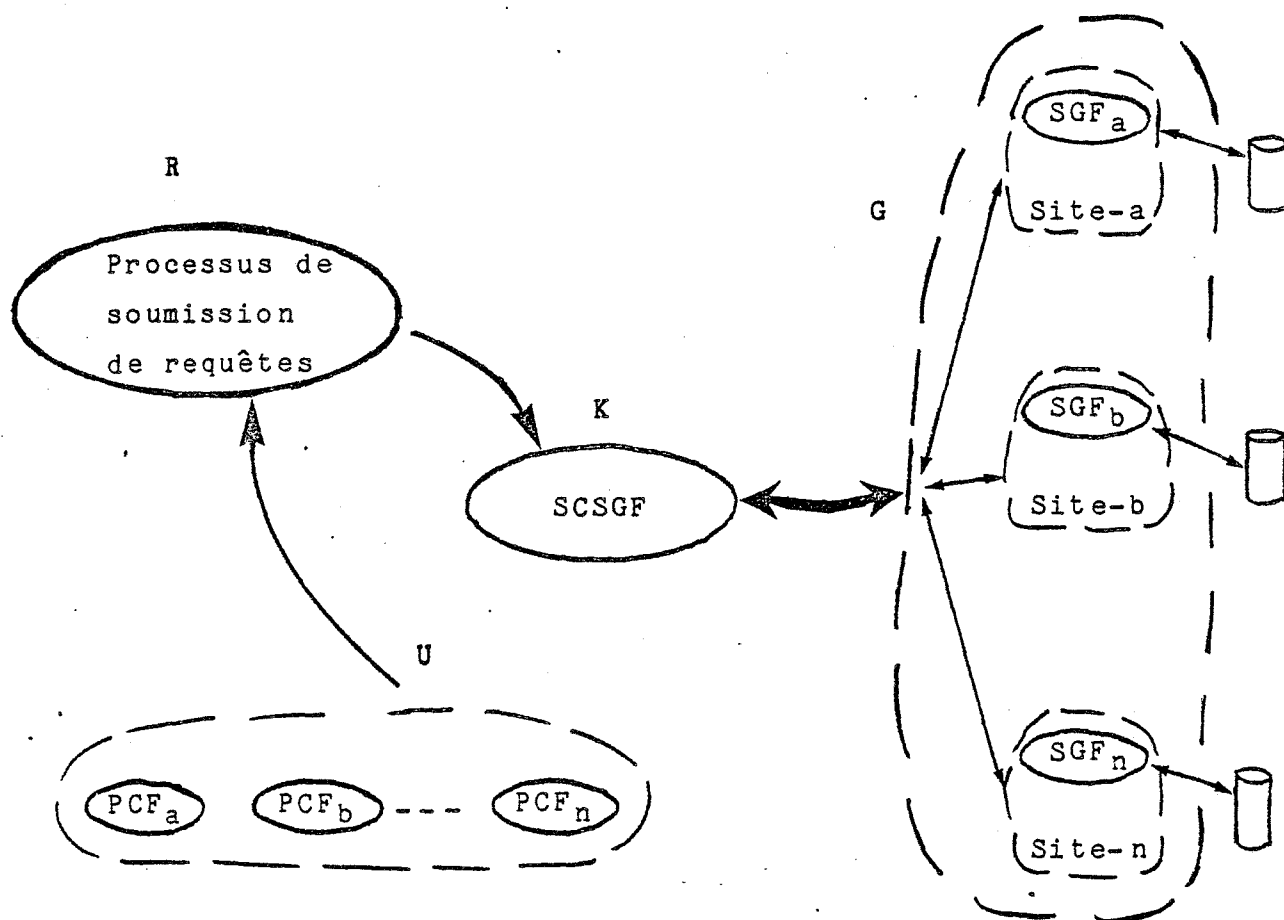
- son existence ou son inexistence;
- son transfert éventuel.

Même si le transfert d'un fichier doit s'effectuer article par article, vu globalement, l'unité de transfert est le fichier tout entier. Pour les accès aux données proprement dites (une fois les transferts des données en entrée effectués), cette méthode est moins pénalisée par le débit des lignes de transmission que la méthode d'accès réseau qui continue à faire des accès à des données distantes pendant le traitement.

2.2. Types de dialogues possibles

L'entité interface qui réalise la coopération entre les différents Systèmes de Gestion de Fichiers est conçue pour être intégrée au niveau de la couche application du modèle de référence (Chapitre IV §3). Ceci suppose l'existence ou le développement en parallèle des autres couches fonctionnelles de niveaux inférieurs. Celles-ci ont pour principale finalité d'offrir une extension aux machines réelles et de servir de bases à une "machine étendue".

C'est donc dans un environnement préexistant que s'intègre l'entité interface. Cet environnement est régi à l'origine par certaines règles d'interactions : en l'absence de l'entité interface, l'entité utilisatrice émet des requêtes au SGF et celui-ci les exécute.



Types de dialogue : U, R
R, K
K, G

SGF : Système de Gestion de Fichiers;
SCSGF: Système de Coopération de SGF;
PCF_i : Producteur-Consommateur de Fichier i,
(Programmes utilisateurs de fichiers).

Figure - 2. Schéma conceptuel logique.

Avec l'introduction de l'entité interface, on veut d'une part apporter le moins de modifications possible au cheminement des requêtes dans un système d'exploitation et,

d'autre part, élargir considérablement le champ d'application des requêtes. Ce dernier objectif est atteint en faisant en sorte qu'un service puisse, fonctionnellement parlant, accéder à un fichier même si ce dernier est localisé sur un site distant. Pour ce faire, il faut à un moment donné :

- connaître les ressources dont aura besoin l'utilisateur ; ce qui pose un problème d'identification et d'analyse anticipée des commandes;
- localiser ces ressources: problème de localisation;
- enfin (dans l'optique adoptée) transférer ces ressources fichiers sur le site d'utilisation: problème d'échange (transfert) d'informations et d'allocation d'espace de stockage.

Par manque d'ouverture des systèmes actuels, toute collaboration entre deux éléments de G (voir figure-1), nécessite la mise en oeuvre de K car les éléments de G s'ignorent et ne peuvent pas communiquer directement.

2.3. Gestion de catalogue

Dans notre approche il est nécessaire de connaître, à un certain niveau du système, la localisation des fichiers afin de déclencher, le cas échéant, les transferts nécessaires vers le lieu d'utilisation du fichier.

De même il est nécessaire d'assurer une unicité des noms de fichiers dans l'ensemble des fichiers accessibles du réseau pour éviter les problèmes d'homonymie. Pour ce faire,

le système de coopération a besoin d'un "catalogue" répertoriant l'ensemble des fichiers accessibles par les services du réseau. Pour la constitution de ce catalogue nous avons envisagé successivement trois solutions:

1) Utilisation des catalogues existants

Chaque Système de Gestion de Fichiers (SGF) d'un site gère par construction son propre catalogue des fichiers du site. Il paraît naturel de penser à réunir logiquement ces différents catalogues en un même ensemble.

L'avantage de cette solution est d'éviter de dupliquer des informations qui existent déjà. Mais cette solution s'avère rapidement très lourde: à chaque demande de fichier il faudra interroger tous les $SCSGF_i$ jusqu'à l'obtention du fichier concerné, s'il existe. De plus, en cas de panne d'un site, par exemple, on ne peut résoudre les problèmes d'homonymie ;

2) Extension du catalogue sur chaque site

Pour pallier les inconvénients de la solution précédente, nous avons envisagé, sur chaque site, de compléter le catalogue standard par un catalogue complémentaire. Ce dernier est un répertoire des fichiers accessibles, externes au site, contenant, entre autres, les noms de ces fichiers et leur localisation.

Dans cette solution chaque site connaît l'emplacement de tous les fichiers, sans effectuer des transactions avec ses partenaires. En contrepartie les mises à jour relatives aux créations ou destructions de fichiers nécessitent des dialogues entre l'ensemble des sites ;

3) Catalogue général

Cette solution est un compromis. Elle consiste en la création d'un catalogue spécifique au système de coopération. Ce Catalogue Général est maintenu sur un des sites; pour des raisons évidentes de fiabilité, il en existe une copie sur un autre site. Chaque site étant autonome et abritant, souvent, tous les fichiers dont ses services ont besoin, l'utilisation du Catalogue Général sera ainsi réduite. En effet, dans la plupart des cas, elle sera restreinte à des événements bien précis tels que:

- création d'un fichier: une consultation du catalogue sera faite afin de résoudre les problèmes d'homonymie, suivie d'une mise à jour;
- destruction d'un fichier;
- récupération d'un fichier distant: lorsqu'un fichier ne se trouve pas sur place le mécanisme chargé de le récupérer s'adressera au Catalogue Général qui l'informera de la localisation du fichier.

Pour des raisons d'optimisation (réduction du nombre de transactions), nous avons choisi les deux sites en fonction du nombre de ressources fichiers qu'ils abritent. Ainsi, lorsqu'un site a besoin d'un fichier distant il est très probable que ce fichier soit localisé sur le site possédant le Catalogue Général et qu'à l'interrogation de ce Catalogue il ait comme réponse le fichier lui-même, et non l'adresse sur un autre site.

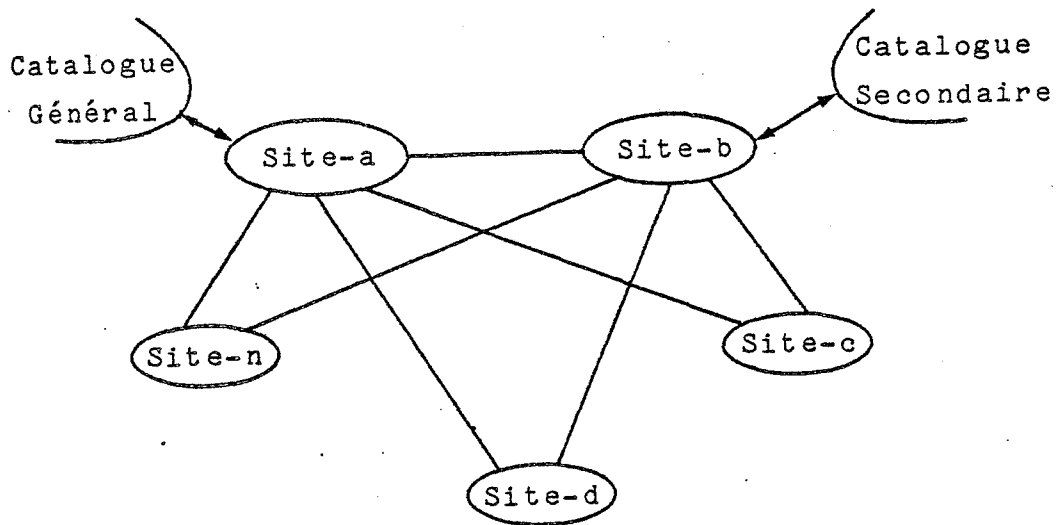


Figure - 3.

La panne d'un site (parmi les machines abritant les services), quel qu'il soit, a peu de répercussions sur le fonctionnement du reste du réseau. De plus, si des pannes simultanées sur plusieurs sites ont lieu, le système continuera à fonctionner d'une façon encore fiable (FRANK, FRISCH: RES-18), (LUNN, BENNETT: GFR-8). En effet, pour le type de contrôle que nous venons de définir, seul le cas d'une combinaison de pannes bien particulière - pannes des deux sites abritant le Catalogue Général et sa copie - peut porter atteinte au bon fonctionnement du système. A cause de la faible probabilité d'un tel événement, cette approche nous est apparue intéressante.

3. Mécanisme de fonctionnement du Système de Coopération

Le but du SCSGF (Système de Coopération de Systèmes de Gestion de Fichiers) est d'assurer une extension potentielle à chaque SGF d'un réseau local, et non de créer une machine spécialisée dans les fichiers (HEBENSTREIT: RES-20). Cette extension est formée par les ressources de l'ensemble des SGF.

Lorsqu'un service a besoin d'un fichier donné, pour optimiser le nombre de transferts réseaux, la recherche de ce fichier sera faite en local, c'est à dire sur le site du service en question. Si le fichier n'est pas rencontré sur place on s'adressera tout de suite au site où se trouve le Catalogue Général des fichiers du réseau. Après une consultation de ce catalogue, trois situations peuvent se présenter, à savoir:

- le fichier est sur place, il se trouve sur le même site que le Catalogue Général. La réponse à la demande de ce fichier sera alors l'envoi du fichier, directement;

- le fichier ne se trouve pas sur place, il est stocké ailleurs. Dans ce cas la réponse à la demande de ce fichier sera l'identification du site où il se trouve. C'est au site qui reçoit cette identification de renvoyer une demande de fichier, cette fois-ci au bon site, en ayant alors comme réponse le fichier, lui-même, s'il est libre;

- le fichier n'existe pas. Dans ce dernier cas l'action prise par la suite dépendra du service qui en a fait la demande.

Les fichiers-services (code objet ou image mémoire), une fois mis en exploitation restent liés au site sur lequel ils ont été créés.

Etant donné que l'allocation des ressources d'un site par un autre ne se produira que si le site demandeur n'en a plus chez lui, le trafic d'information dans le CCR sera ainsi minimisé.

Le fonctionnement du système doit être indépendant du graphe physique. Ce dernier est conçu pour assurer une charge minimale des lignes de transmission (la première interrogation pour la recherche d'un fichier se faisant en local). On pourra envisager la modification du graphe physique suite à des mesures d'évaluation des performances.

3.1. Localisation des fichiers

Les ressources seront allouées sous forme de fichiers sur la demande soit des utilisateurs, soit des services qu'ils utilisent. Ces fichiers peuvent être temporaires ou permanents. Dans un premier temps, pour des raisons de facilité de conception, nous ne manipulons que des fichiers dans leur globalité (l'unité atomique d'accès est le fichier). Mais soucieux de l'extensibilité nous prévoyons la possibilité d'ajouter des couches supplémentaires pour travailler au niveau de l'article.

Afin de minimiser le temps d'attente de l'utilisateur, en réduisant l'"overhead" entraîné par le Système de Coopération, nous avons recherché à stocker le fichier là où il est le plus susceptible d'être utilisé. Cependant, il convient de prévoir, sur chaque site, une zone de travail ("scratch") servant exclusivement au stockage des fichiers temporaires. Les fichiers temporaires sont conservés dans cette zone :

* soit parce qu'ils sont en cours d'utilisation sur ce site (différent de leur site de résidence),

* soit parce qu'ils viennent d'être créés sur ce site dont les capacités de stockage (espace de stockage allouable) sont insuffisantes pour qu'ils puissent y être conservés de façon permanente.

Chaque fichier sera ainsi localisé sur le site où se trouve le service responsable de sa création sauf si, lors de cette action, il n'y a plus de place disponible dans l'espace de stockage allouable de ce site. Dans ce cas, les autres sites du réseau seront interrogés. L'ordre d'interrogation des sites du réseau prend en compte le nombre de ressources qu'ils offrent. Celui qui offre le plus grand nombre de ressources sera interrogé en premier et ainsi de suite.

Or, étant donné que le Catalogue Général des fichiers du réseau se trouve sur le site le plus haut placé dans cette hiérarchie et la copie de ce catalogue sur le site suivant, il est très probable que les fichiers distants soient situés sur un de ces deux sites. Cette répartition des catalogues nous paraît adaptée dans la mesure où on veut réduire au minimum les dialogues entre les sites, lors de la récupération d'un fichier distant (NEGARET: MER-14).

Du point de vue de l'utilisateur, tout se passe comme si son fichier et le service qui l'utilise, résident toujours sur le même site. Pour lui et pour le service qui utilise le fichier, le fait que ce dernier se situe, à un moment donné, sur un autre site ou pas est tout à fait transparent.

3.2. Espace de désignation

Pour manipuler un objet d'un système, l'utilisateur doit le désigner sans ambiguïté. Dans un réseau d'ordinateurs, le mécanisme de désignation doit également permettre d'accéder à des objets ou à des services situés sur d'autres sites (LINDSAY: MER-11).

Pour prévenir les problèmes d'homonymie, il est nécessaire que le nom complet d'un fichier l'identifie de manière non ambiguë; d'où la nécessité de définir l'espace des noms de manière à satisfaire cette contrainte. Le mécanisme de désignation doit également permettre d'accéder à des objets ou à des services situés sur d'autres sites.

Si l'on veut assurer la transparence de la localisation des ressources, la désignation vue par l'utilisateur ne doit pas être affectée par suite d'un simple changement de site du fichier. Cela implique que cette désignation ne doit pas comporter une référence au site de résidence du fichier. Ceci obligerait à changer le nom du fichier dès qu'il changerait de site.

La solution que nous avons adoptée consiste à avoir des noms réseaux universels. On donne à chaque objet du réseau un nom unique indépendant du site de résidence, on résoud à la fois les problèmes d'homonymie et ceux posés par le changement de localisation. Cette méthode a l'avantage d'autoriser la migration d'un objet sans modification du nom réseau universel. La formation de ce nom se fait de la façon suivante :

-

$\langle \text{Nom Universel} \rangle ::= \langle \text{ID. Usager} \rangle \langle \text{ID. Fichier} \rangle$

$\langle \text{ID. Usager} \rangle ::= \langle \text{Nom utilisateur} \rangle | \langle \text{ID. Groupe} \rangle$

$\langle \text{ID. Groupe} \rangle ::= \langle \text{Nom Groupe} \rangle \langle \text{N}^\circ \text{ dans le Groupe} \rangle$

$\langle \text{ID. Fichier} \rangle ::= \langle \text{Nom Fichier} \rangle \langle \text{Type} \rangle \langle \text{N}^\circ \text{ Version} \rangle \dots$

A condition de disposer de plusieurs services à des fins identiques, la méthode a aussi les avantages suivants (CORNAFION: MER-2) :

- elle permet de masquer à l'utilisateur les défaillances d'un système,
- elle facilite une meilleure répartition de la charge sur le réseau,
- elle permet aussi (comme nous venons de le voir) de déplacer un objet d'un site à l'autre à l'insu de l'utilisateur.

A ce problème de désignation, les solutions adoptées sont diverses et passent, souvent, par l'utilisation d'un (ou plusieurs) catalogue(s) (FARBER 72).

3.3. Contrôle de concurrence d'accès

La méthode de contrôle de concurrence du SCSGF s'adresse à un environnement réparti (BADAL: CAC-1) (GARDARIN: CAC-5) (LE LANN: CAC-7, CAC-9) et a pour objectif le contrôle d'accès à des fichiers dans leur globalité. Cette méthode se voulant assez générale, nous utilisons le terme "donnée" (et non fichier) pour ménager de futures extensions.

Dans notre application les sites sont autonomes, et chacun est responsable du contrôle de concurrence chez lui (ROSENKRANTZ et al.: CAC-11). Lorsqu'un utilisateur veut accéder à un fichier F, il émet une requête qui est répercutée sous forme de commande jusqu'au SGF (Système de Gestion de Fichiers) du site auquel il est connecté, à la suite de quoi deux cas sont possibles :

1) Le fichier F est situé sur le même site que l'application : si le fichier est libre, l'utilisateur l'aura tout de suite à sa disposition ; si non, le SGF l'informerait que le fichier F est occupé, comme il aurait fait normalement s'il s'agissait d'un site indépendant et isolé.

2) Le fichier F est sur un site autre que celui de l'application de l'utilisateur : la commande de demande du fichier F est donc répercutée au site abritant le Catalogue Général, de façon transparente à l'utilisateur. Si le fichier recherché se trouve sur le même site que le Catalogue Général il est envoyé temporairement au site demandeur, s'il est libre. Dans le cas où il n'est pas libre, pour l'utilisateur, c'est son SGF local qui l'informerait de la non disponibilité du fichier. Si le fichier n'est pas sur le même site que le Catalogue Général, son adresse est envoyée au SCSGF_i (composant du SCSGF) du site de l'application qui s'adressera au site abritant le fichier F, comme dans le cas d'un fichier qui se situe sur le même site que le Catalogue Général.

Le Catalogue Général ne contrôle pas la disponibilité des fichiers; son rôle est de répertorier ceux qui existent dans le réseau, et leur localisation respective. Lorsqu'un fichier est envoyé temporairement sur un autre

site, il y restera jusqu'à sa libération par l'utilisateur, et sera alors rapatrié sur son site d'origine. Entre temps, si un autre utilisateur essaie d'accéder à ce fichier, c'est au site qui l'abrite en permanence qu'arrivera sa demande. Celui-là lui répondra "occupé", tant que le fichier n'est pas de retour c'est à dire, libéré. En fait, tout se passe comme si le fichier sollicité, était en cours d'utilisation localement. Le contrôle de la concurrence d'accès est donc réparti entre tous les sites du réseau, en ce sens que chaque site est responsable de l'utilisation des fichiers qu'il possède, même lorsque ceux-ci sont en cours d'utilisation ailleurs.

Dans le cas précis du Centre de Calcul Réparti, la cohérence interne (BERNSTEIN et al.: CAC-2) (THOMAS: CAC-15) au niveau de chaque site, ainsi que la protection (c'est à dire, l'attribution et le contrôle de droits d'accès différents selon le type d'utilisateur), sont assurées par construction : chaque site en étant responsable, le fait à sa manière et suivant la technique qui lui est propre.

Le CCR n'a pas, jusqu'à présent, des données redondantes. Dans son cas précis donc, assurer la cohérence globale des données revient à l'assurer pour les fichiers utilisés en dehors du site d'origine c'est à dire, pour ceux qui sont temporairement dupliqués (JUREMA: CAC-6). Ceci est fait par le verrouillage du fichier sur son site d'origine et ensuite son remplacement par sa copie, lorsque celle-ci, après sa mise à jour, aura été libérée et rapatriée (ESWARAN et al.: CAC-4).

Lorsqu'une panne survient sur un site abritant un fichier qui a une copie en cours d'utilisation sur un autre site, tout se passera comme si le fichier était utilisé localement. A la fin de l'utilisation de la copie du fichier

son rapatriement ne sera pas effectué, même si le site est à nouveau en service, la copie sera détruite et l'utilisateur recevra un message lui communiquant l'incident. En fait, les informations concernant l'envoi de la copie du fichier sont perdues au moment de la panne du site abritant l'original. Tout se passe donc, comme si la panne survenait sur le site où tourne l'application.

Le choix du site sur lequel seront créés les fichiers des utilisateurs est fait de manière à minimiser la création de copies (FISCHER et al.: MER-6) (MAHMOUD, RIORDON: MER-13). Toutefois, si cela arrive trop souvent pour un fichier donné, il pourra changer définitivement de site. C'est pour cela que les pannes éventuelles d'un site n'auront généralement pas de répercussions importantes.

En effet, notre procédure de reprise consiste à faire en sorte que, d'une part aucun site n'accepte le rapatriement d'un fichier pour lequel il n'a pas toutes les informations concernant son envoi et, d'autre part que l'utilisateur soit, le plus tôt possible, au courant de la panne. Ainsi, lorsqu'un site réintègre le système il n'y a pas le risque de création d'une "deuxième copie" pour un même fichier. La situation est donc facilement contrôlable, et de façon simple.

4. Le Système de Coopération proposé

4.1. Caractéristiques

Le SCSGF a comme caractéristique principale son indépendance vis-à-vis des systèmes entrant en jeu. Il devrait répondre à priori au besoin exclusif du CCR (Centre de Calcul Réparti). L'étude initiale que nous avons

effectuée a éveillé l'intérêt porté à ce sujet. Ceci est ressorti en fait, du nombre restreint de publications dans ce domaine, par la place donnée à ces publications dans les manifestations et revues les plus importantes et, à fortiori, par l'extension de son champ d'application.

Cela a constitué donc des raisons suffisantes pour élargir les concepts sur lesquels serait fondé le SCSGF. Dès lors, il s'agissait de concevoir un système indépendant et ouvert. Indépendant des systèmes entrant en jeu, il n'était plus question d'intégrer un environnement particulier (celui du CCR), et ouvert à des évolutions éventuelles du SCSGF lui même ou de son environnement.

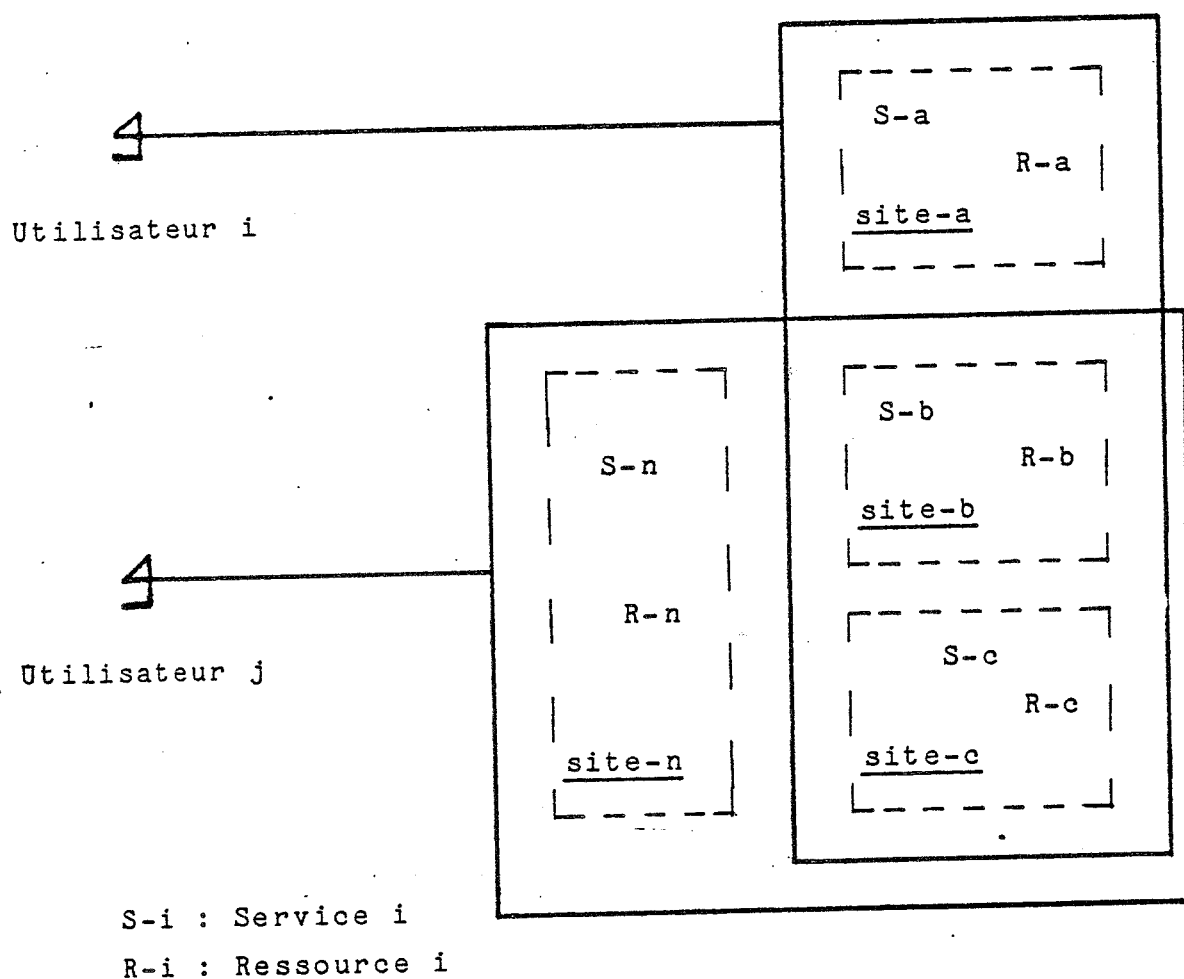


Figure - 4. Vision offerte à l'utilisateur.

Le SCSGF se propose, quel que soit son environnement, de remplir ses fonctions de façon transparente à l'utilisateur. Celui-ci a une vision qui lui est propre du système dont il dispose. Le SCSGF offre donc un environnement personnalisé à chaque utilisateur, qui par ailleurs l'imagine sur un site unique. Cela lui permet, au cours de sa session de travail, d'utiliser des services ou ressources situées sur différents sites du réseau local, comme s'ils appartenaient à un seul.

La fonction de coopération entre SGF fait que le SCSGF est vu plus ou moins comme un "Gestionnaire de Fichiers Réparti". Toutefois, sa vocation première est surtout de gérer, avec l'aide de chacun des SGF appartenant aux différents sites, l'espace de stockage global du réseau qui le supporte.

Un éventuel manque de place (saturation) survenu sur un site donné peut être compensé par une certaine disponibilité sur un autre site. Ce n'est pas forcément le nouveau fichier qui sera situé ailleurs car les incompatibilités existant entre les ordinateurs d'origines différentes limitent ces transferts, dans certains cas, à des types restreints de fichiers. Le cas échéant, un fichier créé depuis longtemps pourra être envoyé sur un autre site afin de libérer de la place localement pour un nouveau fichier, si la structure de celui-ci empêche son envoi. Un codage des différents types de structures existantes peut supprimer ce problème. Ceci enrichit le SCSGF d'une fonction de plus, celle de traducteur automatique de structure.

4.2. Méthodologie

L'architecture conceptuelle du SCSGF est indépendante de son schéma physique. Cette indépendance donne au SCSGF la souplesse suffisante à son adaptation à des systèmes très divers. Sa localisation physique est fonction de l'environnement-système qui l'accueille. Elle peut être concentrée sur un site unique, pour un système centralisé, réparti sur chaque site, pour un système réparti, ou encore avoir toute autre configuration intermédiaire, pour les systèmes mixtes. Dans tous les cas de figure l'important est que le $SCSGF_i$ doit être situé à un niveau tel qu'il soit capable d'intercepter les commandes envoyées, au système d'exploitation local, par les producteurs-consommateurs de fichiers.

Le $SCSGF_i$ doit occasionnellement interagir avec ses homologues ($SCSGF_{j,k,\dots}$), via le Catalogue Général. A cette interaction généralement est associé un dialogue direct avec le SGF_i . Après l'interception et l'analyse des commandes relatives aux fichiers, envoyées à l' OS_i (système d'exploitation du site i), le $SCSGF_i$ dialogue avec le SGF_i afin de s'assurer que celui-ci possède tous les éléments référencés par la commande. Si ce n'est pas le cas, avant de libérer la commande vers l' OS_i , le $SCSGF_i$ interagira avec ses homologues dans le but de composer localement l'environnement nécessaire à la bonne exécution de la commande. Les interactions entre $SCSGF_{i,j,\dots,n}$ sont nécessaires lors d'une création ou destruction d'un fichier ou l'accès à un fichier distant.

Le SCSGF est donc applicable à tout système disposant d'un moyen de communication (couches de niveau 3 et 4 du modèle ISO et celles de niveau inférieur) fiable entre les différents sites et qui offre la possibilité de "surveiller" le système d'exploitation (OS) local à chacun des sites. Cette surveillance s'avère importante pour:

- l'interception et l'analyse des commandes, relatives aux fichiers, envoyées à l'OS;
- la récupération éventuelle d'un fichier distant, afin de construire localement l'environnement du service devant être activé;
- la transparence à l'utilisateur de la localisation des ressources auxquelles il a accès.

La localisation, dans le système, des composants du SCSGF doit être telle qu'il ait la possibilité de simuler à l'utilisateur la présence constante sur le site de toutes les ressources qui lui sont offertes. Sa localisation donc est dépendante uniquement des caractéristiques du système qui l'accueille (centralisé, réparti, etc.).

4.3. Apport du SCSGF

Après la phase initiale de développement des "réseaux informatiques", la deuxième phase a vu l'attention qui leur était dédiée, partagée entre eux et les applications. C'est ainsi que quelques travaux, ayant pour but de supprimer les incompatibilités provoquées par l'hétérogénéité des ordinateurs mis en relation, ont été

encouragés. Pour la plupart, ces travaux consistent à rajouter au système existant, une couche logicielle qui peut être orientée utilisateur ou service.

Lorsque cette couche logicielle rajoutée au système est orientée utilisateurs, elle est chargée de donner à ceux-ci une vision globale et unique des couches de niveaux inférieurs. Les particularismes de chaque système seront filtrés et l'utilisateur n'aura connaissance que du tronc commun.

Si cette couche logicielle rajoutée au système est orientée service, le plus courant est de simuler au niveau de chaque service les caractéristiques nécessaires à son déroulement normal. Ceci est fait, éventuellement au moyen d'une traduction de structure, indépendamment du lieu d'où proviennent les entités dont le service a besoin.

Les intérêts, avantages et inconvénients de chacune de ces orientations, ainsi que les degrés de difficulté "d'implémentation", varient avec l'environnement.

CHAPITRE VII

ARCHITECTURE RESULTANTE DU SYSTEME DE COOPERATION

1. Fonctionnalité du système de coopération
2. Décomposition client-serveur
3. Schéma de principe
4. Les entités fonctionnelles
 - 4.1. Le processus de soumission de requêtes
 - 4.2. L'entité client
 - 4.3. L'entité serveur
 - 4.4. Le gestionnaire de catalogue
5. Structure et gestion de catalogue
6. Approche fonctionnelle
 - 6.1. Architecture distribuée du système de coopération de SGF
 - 6.2. Environnement d'un sous-système local
7. Interactions entre les différentes entités

1. Fonctionnalité du système de coopération

Le système de coopération fournit un service de gestion de fichiers global qui intègre en son sein les possibilités des systèmes de gestion de fichiers locaux et offre à ces derniers une ouverture sur l'extérieur.

Un système de gestion de fichiers réseau peut être considéré comme étant composé de deux sortes de composants (GIEN: MAS-9), (DONNELLEY: MAS-6) :

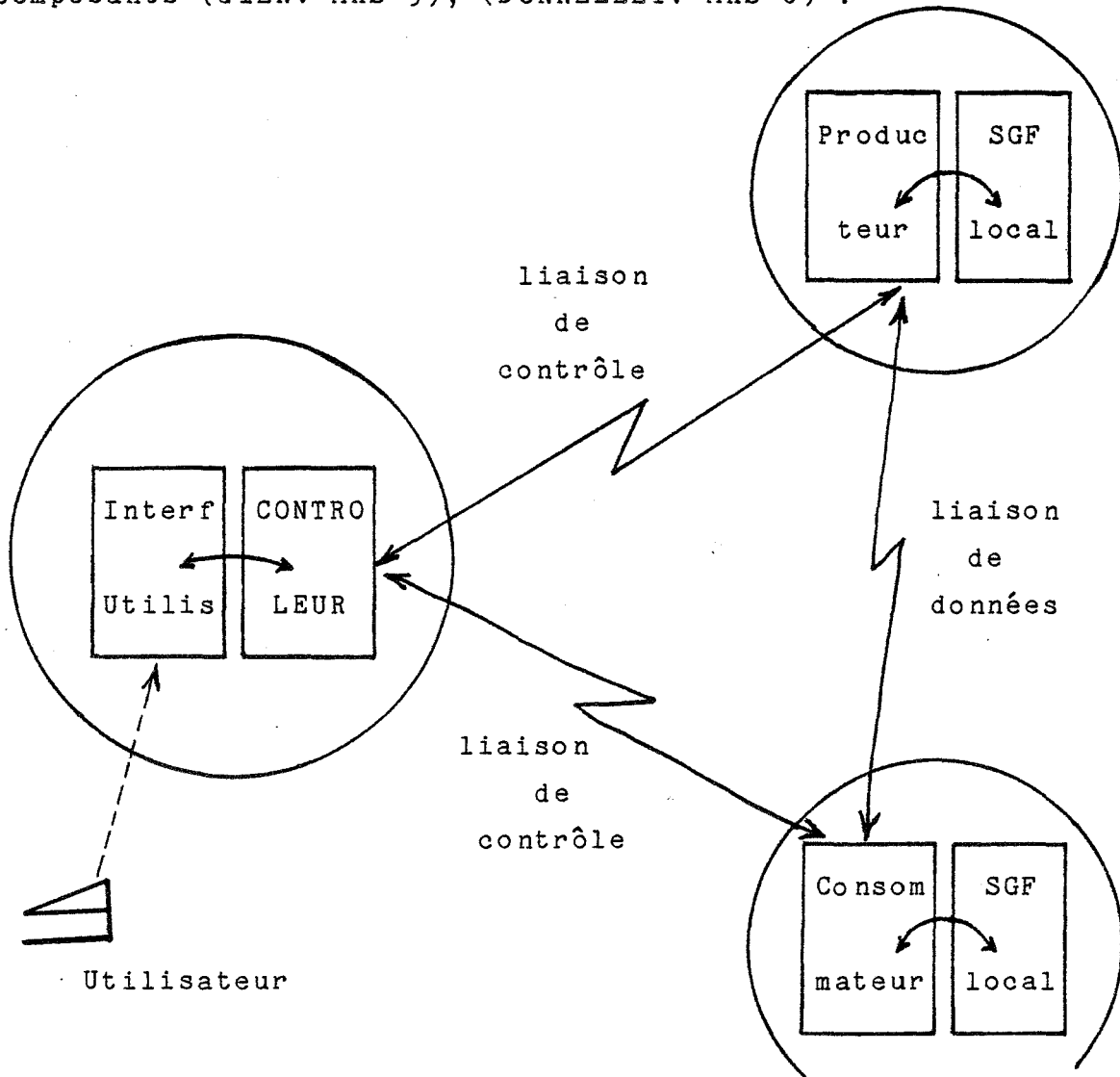


Figure - 1. Composants d'un système de gestion de fichiers réseau

- des points de service offrant des accès au système de gestion de fichier local;
- des points de contrôle qui supervisent et synchronisent les différentes actions des serveurs fichiers.

Dans cette optique, un transfert de fichiers met en jeu un contrôleur et deux serveurs (un serveur situé sur le site émetteur, c'est le producteur et un serveur situé sur le site destinataire : serveur consommateur). Pour mener à bien ce transfert, un dialogue s'établit entre ces trois entités.

Dans le cas où le système réparti repose sur un seul contrôleur, le principal reproche qu'on peut faire à ce modèle est que le contrôle est centralisé. En effet, il est confié à l'unique contrôleur.

En plus du goulot d'étranglement que représente un tel contrôleur, cette solution est très peu fiable et n'offre pas de résistance aux pannes du site abritant le contrôleur central.

C'est pour toutes ces raisons que nous nous sommes orientés vers un système à contrôle réparti laissant le maximum d'autonomie aux sous-systèmes participants.

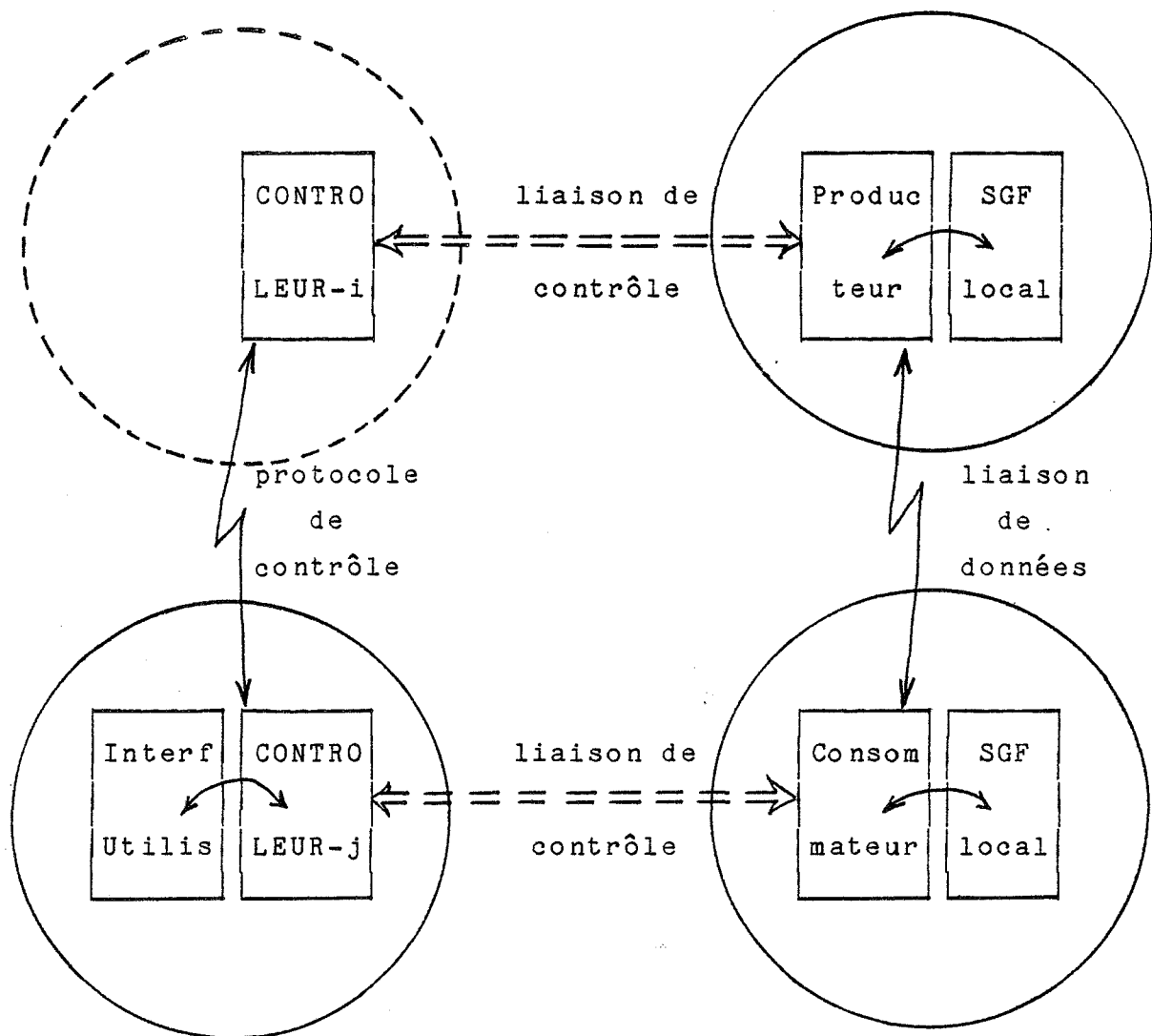


Figure - 2. Alternative à contrôle réparti

2. Décomposition client-serveur

Sans vouloir pour l'instant rentrer dans les détails de réalisation d'un sous-système fonctionnel, nous pouvons distinguer les entités principales suivantes dans le système de coopération :

- une partie du système de coopération à laquelle les enveloppes des services soumettent leurs requêtes de localisation des fichiers; cette partie dialogue avec les autres composants du système de coopération,

- une partie plus orientée vers l'environnement local. Cette partie traduit les requêtes du système de coopération en ordres propres au SGF standard du constructeur. Cette partie joue donc le rôle d'un serveur local.

Cette première décomposition obéit tout à fait au schéma client-serveur.

Le Système de Coopération de SGF est donc constitué de deux sous-entités:

- une sous-entité client qui interprète les commandes des utilisateurs, dialogue avec l'environnement local et l'environnement distant selon le cas,

- une entité serveur composée d'un ensemble de voies d'accès au Système de Gestion de Fichiers local. Cette sous-entité serveur joue le rôle d'une interface entre le Système de Coopération et le SGF local. Elle fonctionne comme un serveur virtuel car les voies d'accès qu'elle fournit peuvent être considérées comme autant de services.

A ces deux entités, il convient d'ajouter :

- l'entité gestionnaire de catalogue dont le rôle est de gérer le catalogue de l'ensemble des fichiers accessibles du réseau.

Chacune de ces entités peut-être centralisée ou répartie.

3. Schéma de principe d'une architecture

Pour illustrer le modèle d'architecture, prenons l'exemple d'une requête et suivons son cheminement à l'intérieur du système de coopération.

Le processus de soumission de requêtes s'adresse au client qui joue le rôle d'un contrôleur global. Il lui indique l'action à accomplir ainsi que l'identification du fichier concerné.

Le client consulte le gestionnaire de catalogue pour que ce dernier lui fournisse la localisation du fichier.

Ensuite le client s'adresse aux deux serveurs (producteur et consommateur) pour leur faire exécuter l'action nécessaire.

A la fin de l'action, le client est averti, et il retourne alors un compte rendu au processus de soumission de requêtes qui à son tour le transmet au niveau supérieur.

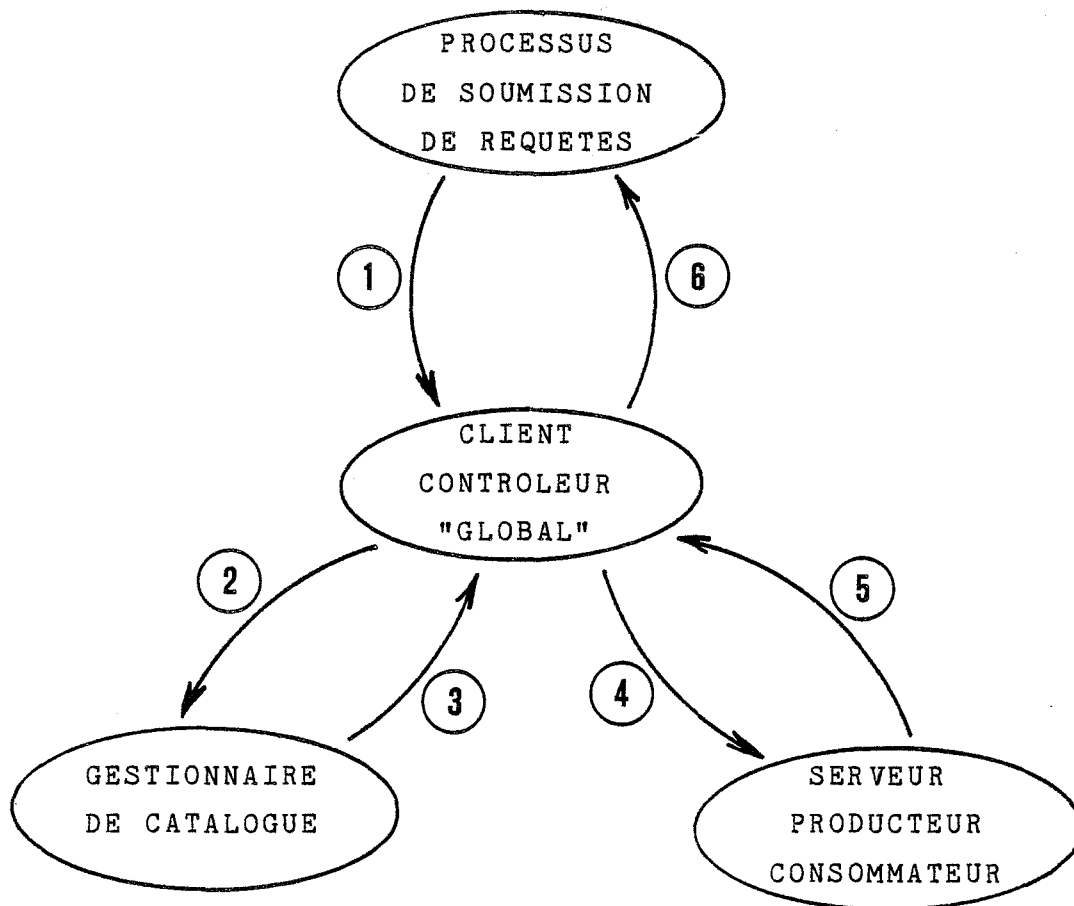


Figure - 3. Schéma de principe.

4. Les entités fonctionnelles

Nous allons étudier le rôle de chacune des entités fonctionnelles qui composent le système de coopération.

4.1. Le processus de soumission de requêtes

L'utilisateur du Centre de Calcul Réparti émet des requêtes qui sont analysées par l'interprète du

langage de commande. C'est une partie de ce dernier qui fait la reconnaissance des ressources fichiers. Il s'adresse alors au Système de Coopération à travers le processus de soumission de requêtes.

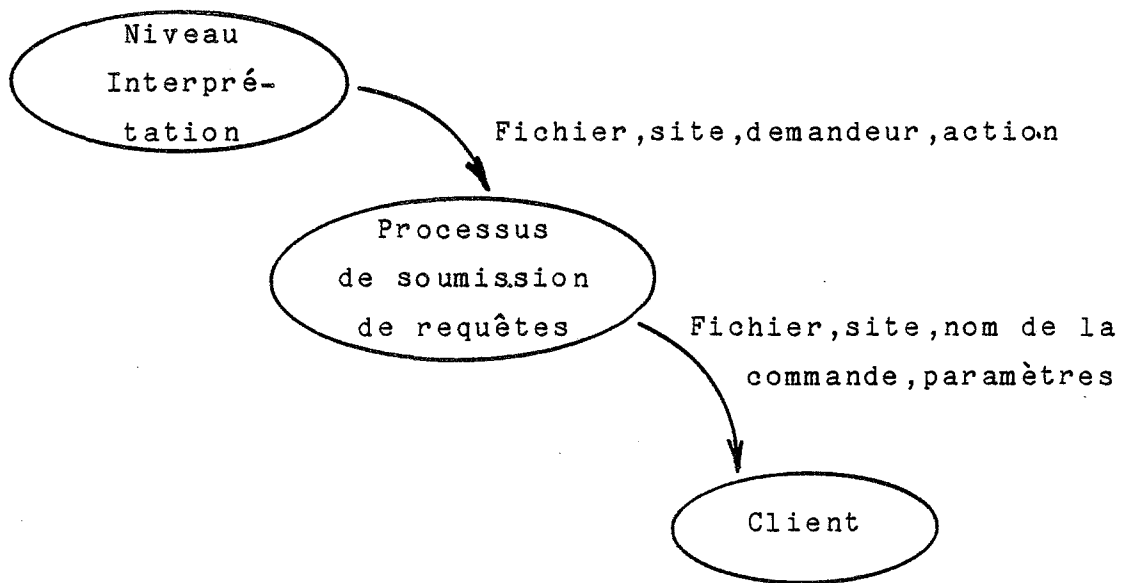


Figure - 4. Processus de soumission de requêtes.

Des requêtes sur les fichiers peuvent provenir directement de programmes utilisateurs ou être des commandes opérateurs ou encore elles peuvent résulter de l'exécution de procédures cataloguées.

Pour rendre homogène la façon dont les requêtes sont soumises au système de coopération, nous avons distingué une entité particulière qui interprète les différentes requêtes fichiers et les soumet au système de coopération. Cette entité sert en fait d'interface avec le système de coopération et elle permet d'avoir une vision globale de l'ensemble des fichiers accessibles du système.

4.2. L'entité client

C'est le composant central du système de coopération. Il centralise toutes les requêtes relatives aux fichiers réseaux. Dans une première phase il gère des requêtes qui manipulent le fichier dans sa globalité. Ces requêtes concernent la création, la destruction ou le changement de localisation (transfert).

Le client prend contact avec le serveur récepteur et le serveur émetteur pour faire exécuter les requêtes.

4.3. L'entité serveur

Chaque entité serveur peut être producteur ou consommateur de fichier.

Chaque serveur doit :

- être capable de s'assurer qu'un fichier donné est bien chez lui,
- pouvoir à la demande émettre un fichier à destination d'un autre site,
- être en mesure de recevoir un fichier en provenance d'un autre site,
- posséder un mécanisme de supervision et de reprise en cas d'erreur au cours du transfert.

Cette dernière condition nécessite la gestion de points de reprises dans le dialogue entre deux serveurs distants.

Le transfert d'un fichier nécessite :

- d'une part un dialogue entre le client et chacun des deux serveurs ;
- d'autre part un dialogue entre les deux serveurs.

* On peut considérer que c'est le serveur récepteur qui prend l'initiative du dialogue et demande à l'émetteur de lui envoyer le fichier. Ceci suppose qu'il est prêt à le recevoir et que les problèmes d'allocation de ressources sont déjà résolus au niveau local au moment où cette demande est émise.

* On pourrait envisager que c'est le client qui prend contact avec les deux serveurs pour les mettre d'accord sur les modalités du transfert.

A la fin du transfert le mécanisme de supervision doit prévoir la mise à jour du catalogue. En effet comme il y a eu changement de localisation et que celle-ci est un des paramètres stockés dans le catalogue, il faut le modifier. Il faut en plus décider de la rupture de la liaison et avertir le demandeur de la fin de la requête.

4.4. Le gestionnaire de catalogue

Il reçoit du client des interrogations concernant la localisation d'un fichier et des requêtes devant entraîner la mise à jour du catalogue. Ces commandes sont du genre :

- donner la localisation de tel fichier
- créer le fichier

- détruire le fichier
- lister ou changer les paramètres d'un ou de plusieurs fichiers.

5. Structure et gestion du catalogue

Le catalogue est un répertoire des fichiers accessibles du réseau. Chaque fichier répertorié a un certain nombre de caractéristiques statiques qui lui sont associées :

- le nom réseau universel (pouvant inclure certains des attributs qui suivent),
- le nom local éventuel donné par l'utilisateur,
- le type du fichier (source, objet, image)
- le site de résidence,
- le volume de résidence (éventuel),
- la taille du fichier,
- l'organisation physique,
- les méthodes d'accès possibles,
- les droits d'accès et les protections...

Le catalogue sera stocké sous forme de fichier. Cependant il semble intéressant de le décomposer en une partie primaire et une partie secondaire.

Sur la partie primaire du catalogue, on met les noms de fichiers ainsi que les sites de résidence de chacun et un pointeur sur l'entrée correspondante de la partie secondaire.

Sur la partie secondaire, on met les autres caractéristiques de chaque fichier. Ce découpage permet de réduire la taille de la partie primaire et d'accélérer ainsi les recherches de localisation.

6. Approche fonctionnelle

Après avoir distingué les différentes entités fonctionnelles qui composent l'application, nous allons en faire une approche plus précise en dégagant l'architecture de chaque entité et les éléments qui composent son environnement.

La décomposition fonctionnelle donne l'occasion de faire une ébauche de quelques-uns des aspects fonctionnels d'un Sous-Système de Coopération (composant du Système de Coopération sur un site).

L'application SCSGF, étant une application de niveau utilisateur, suppose l'existence des niveaux fonctionnels inférieurs qui permettent d'offrir des extensions aux machines réelles.

Ces niveaux fonctionnels inférieurs ont déjà été analysés dans le chapitre "Mécanismes d'exécution répartie". Sur chaque site, nous supposons l'existence d'un point d'accès à un service de transport et d'un ensemble de conventions (ou protocoles) permettant d'utiliser au mieux ce service.

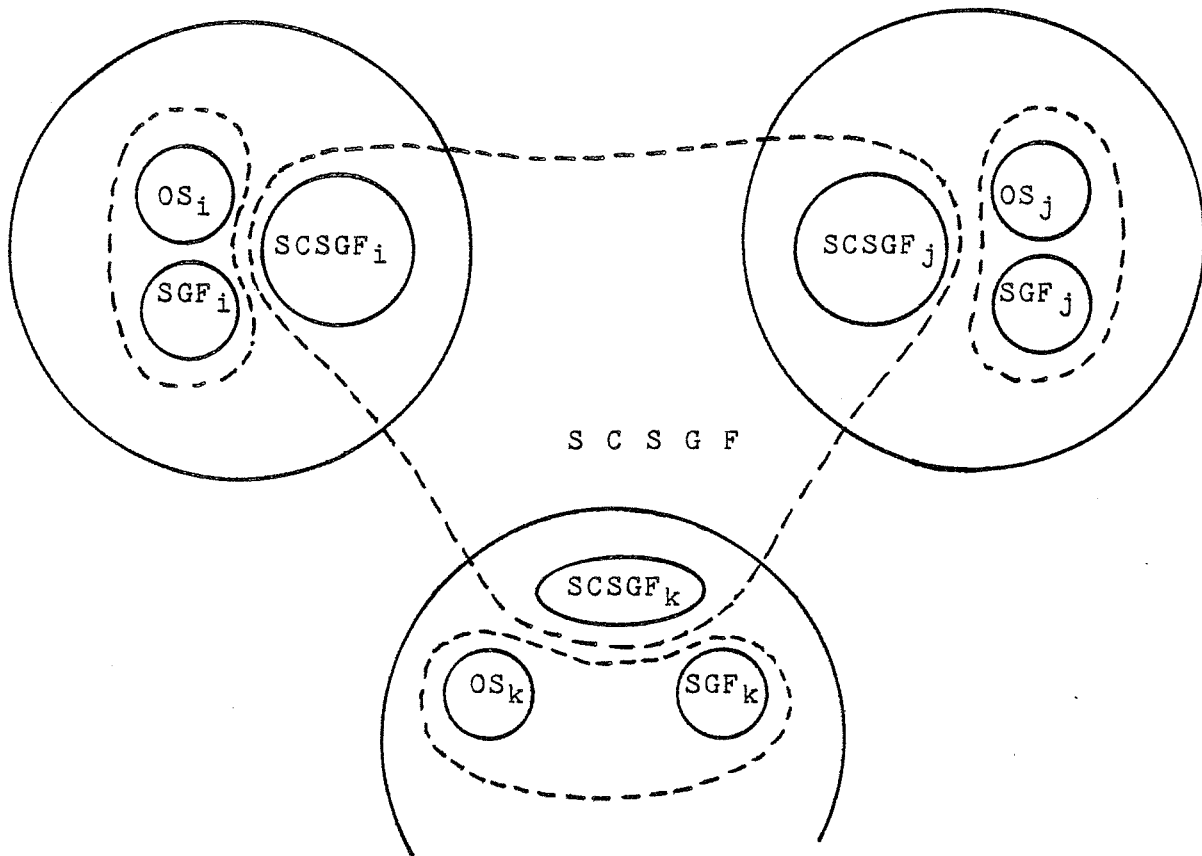
6.1. Architecture distribuée du système de coopération

L'application Système de Coopération de Systèmes de Gestion de Fichiers (SCSGF) est composée d'un ensemble de sous-systèmes fonctionnels géographiquement distribués.

Par la suite nous appellerons:

- SCSGF la vision globale qu'on a de l'application,

- $SCSGF_i$ le composant fonctionnel de l'application implanté sur le site i .



SCSGF : Système de coopération de SGF

SGF : Système de Gestion de Fichiers standard

OS : Système d'exploitation

Figure - 5. Architecture distribuée du Système de Coopération de SGF (SCSGF).

6.2. Environnement d'un sous-système local

Chaque $SCSGF_i$ (Sous-Système de Coopération de SGF) implanté sur un site a dans son environnement:

- un Système d'Exploitation (ou partie de celui-ci) OS_i ;
- un Système de Gestion de Fichiers SGF_i (celui du constructeur,
- un point d'accès à un service de transport lui permettant de dialoguer avec les autres $SCSGF_j$.

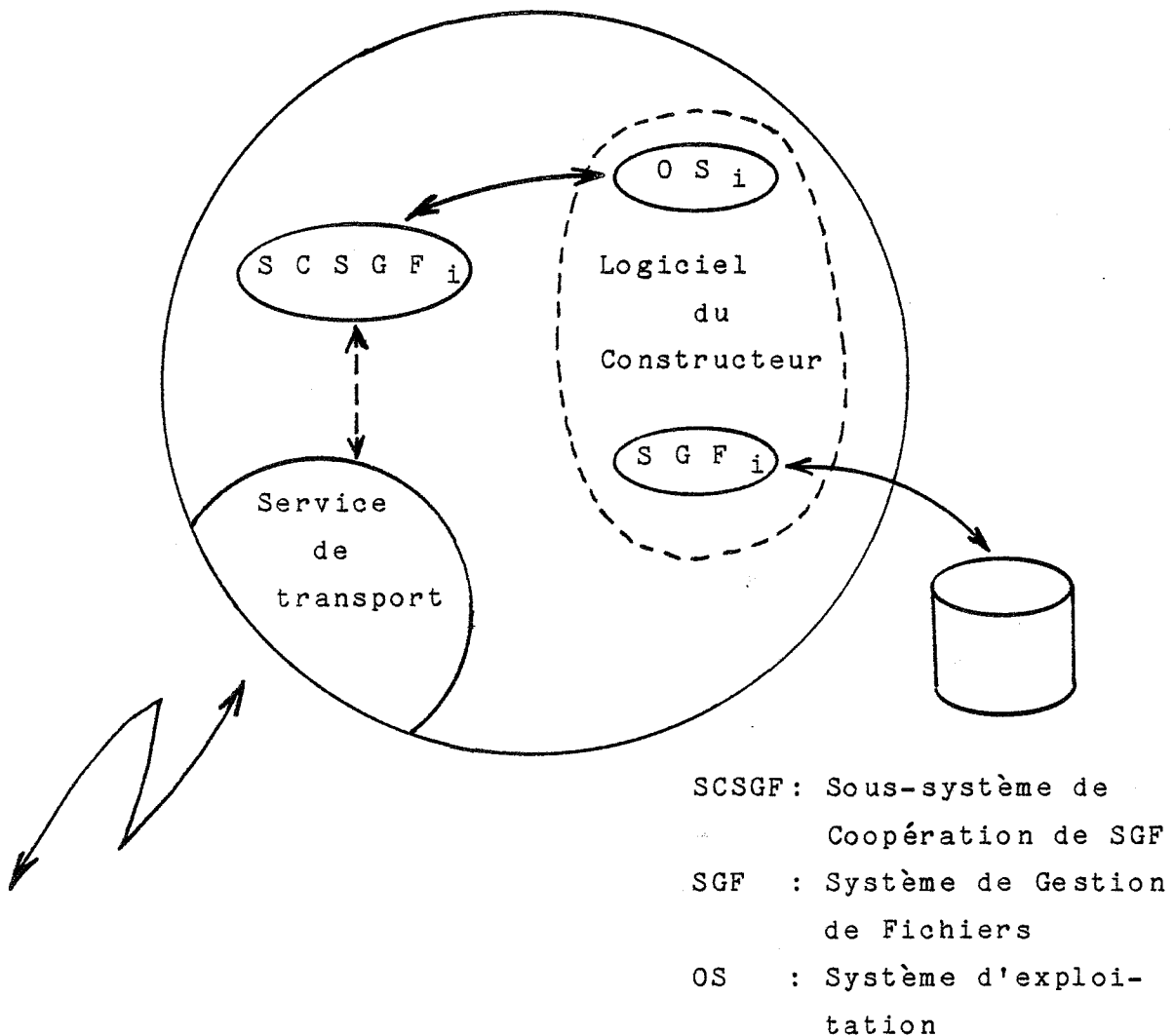


Figure - 6. Environnement d'un Sous-Système de Coopération de SGF.

7. Interactions entre les différentes entités

Après avoir défini les différentes entités fonctionnelles qui composent le Système de Coopération, nous allons mettre en évidence les interactions qui existent entre elles.

Chaque groupe d'actions d'une entité correspond à différents états qu'on pourrait modéliser par un automate d'états finis. Mais nous préférons à cette représentation celle des interactions qui traduit mieux les interdépendances fonctionnelles qui existent entre les entités.

Pour ne pas compliquer inutilement les schémas,

- la répartition de l'entité client,
- la duplication du catalogue et des entités gestionnaires associées,

ne seront pas représentées. Nous considérons que c'est à chacune de ces entités de gérer sa propre répartition en utilisant l'infrastructure fournie par le réseau de transport et les moyens de communication interprocessus.

Les différentes actions sont :

- demande de destruction,
- demande de création,
- demande de récupération d'un fichier.

A chacune d'entre elles correspond un compte rendu approprié.

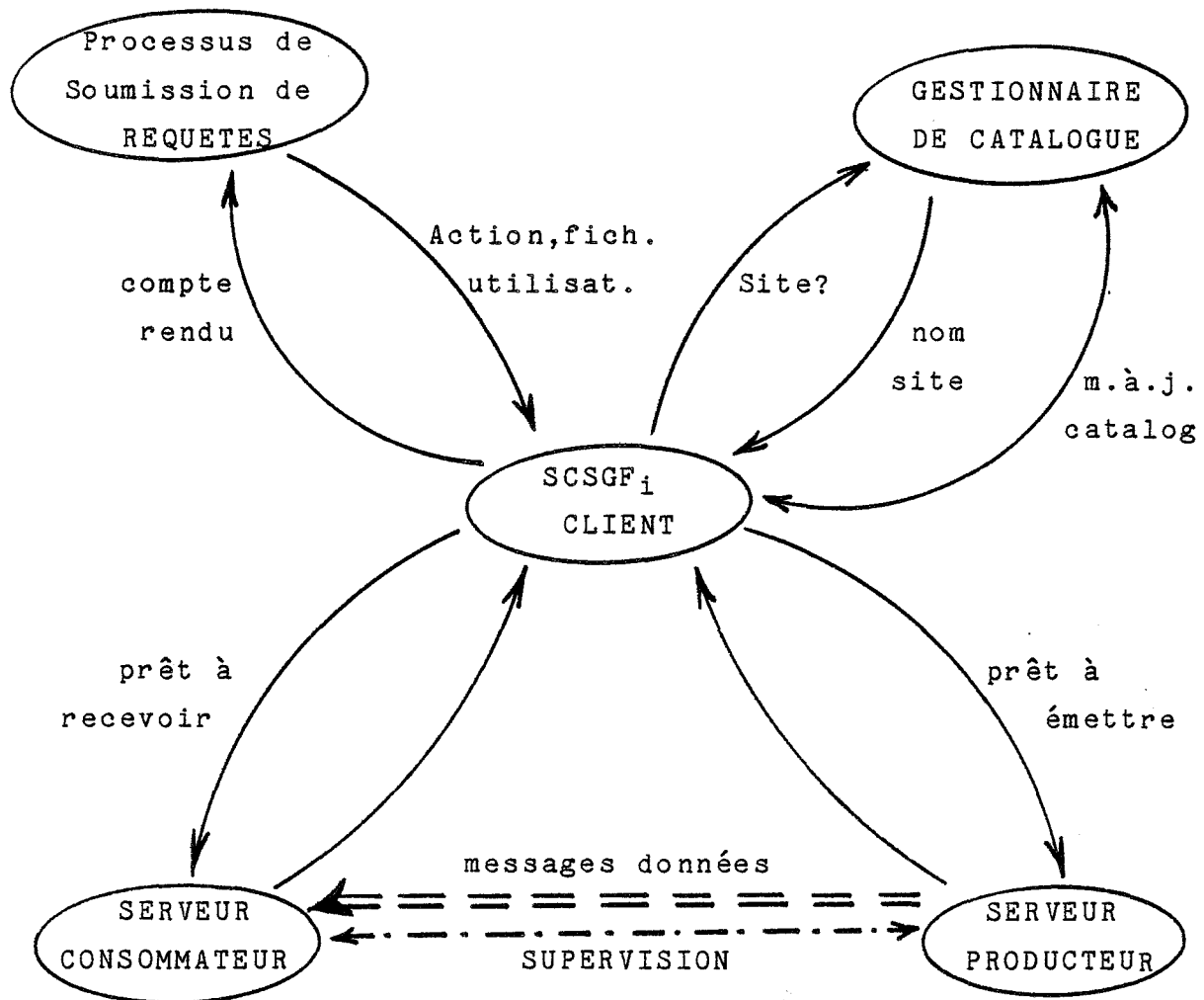


Figure - 7. Interactions entre entités.

Après avoir :

- d'une part analysé la structure et le fonctionnement du CCR, ses mécanismes généraux,
- d'autre part défini une architecture pour le Système de Coopération de Gestion de Fichiers,

il nous reste à examiner l'articulation de ces deux architectures et le principe des interactions fonctionnelles existant entre les entités de l'une et de l'autre.

CHAPITRE VIII

LE CENTRE DE CALCUL REPARTI ET SON OUTIL SCSGF

1. Le projet CCR
 - 1.1. Idées directrices
 - 1.2. Le correspondant
 - 1.3. Les moyens de communication
 - 1.4. Approche fonctionnelle
 - 1.5. La fonction d'interprétation
2. Interaction entre le CCR et le SCSGF
 - 2.1. Localisation du SCSGF dans le CCR
 - 2.2. Exemple d'utilisation possible
3. Extension à d'autres systèmes

1. Le projet CCR

1.1. Idées directrices

La conception de l'architecture du CCR (Centre de Calcul Réparti) a été guidée par un certain nombre d'idées directrices dont l'une des plus importantes est l'approche fonctionnelle qui consiste à :

- distinguer les entités caractérisées chacune par sa fonction ,
- et à déterminer ensuite les interactions entre elles (CART: RES-5).

Cette approche fonctionnelle trouve sa justification, à postériori, dans le fait que les entités fonctionnelles composant le CCR sont définies dans un environnement réparti. De plus, comme elles peuvent elles-mêmes être réparties, toute hypothèse préalable quant à leur localisation conduirait à poser mal le problème de leur réalisation.

Chacune de ces entités doit disposer d'un maximum d'autonomie pour autoriser un déroulement en parallèle de certaines activités avec, au besoin, une synchronisation. Ces entités sont au nombre de trois :

- les utilisateurs,
- le correspondant,
- les services réseaux.

Chaque mini-ordinateur participant conservant les caractéristiques qui lui sont propres (en particulier le système d'exploitation - OS), ceci permet: un fonctionnement en dégradé (grâce à l'autonomie des minis) en cas de panne du réseau, et de bénéficier des évolutions du système apportées par le constructeur.

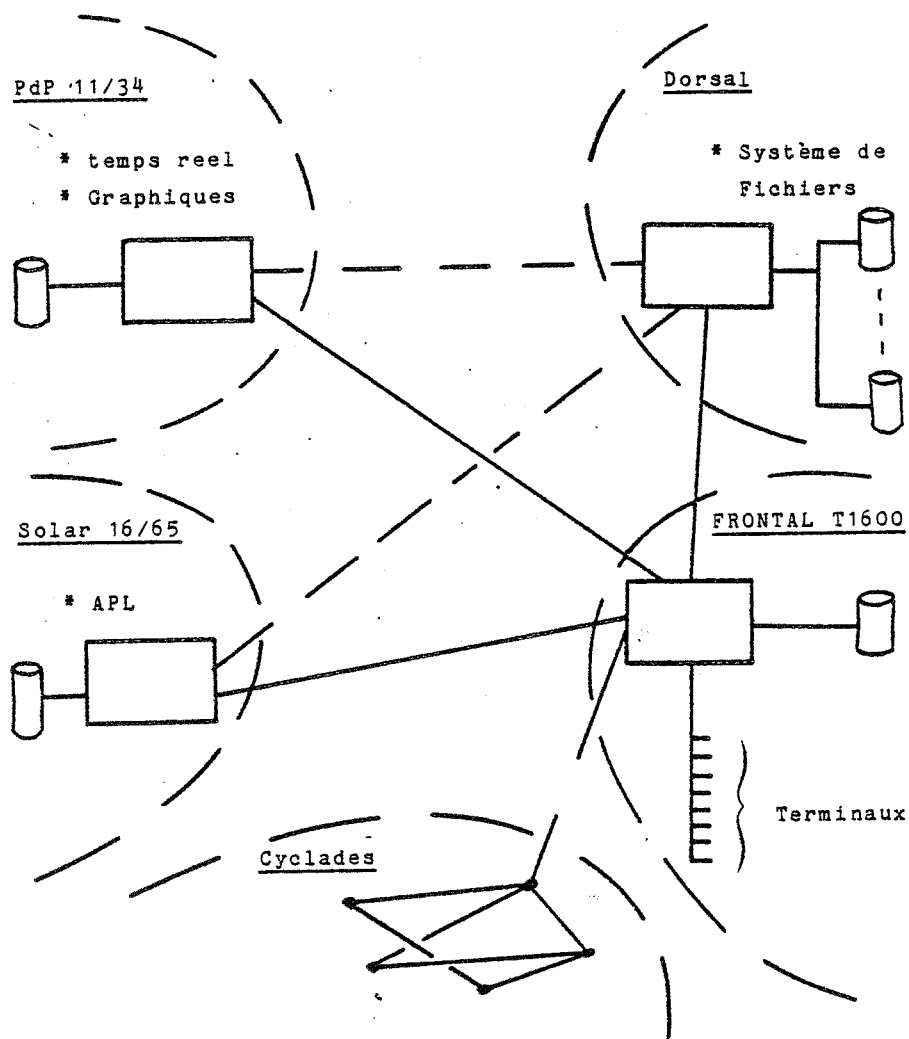


Figure - 1.

1.2. Le correspondant

Le Frontal est chargé de l'aiguillage entre tous les éléments du CCR. Il a une vision globale du système, il se divise en deux parties principales: Correspondant-Utilisateur, à qui les utilisateurs s'adressent, et Correspondant-Service, à qui les services s'adressent.

Le correspondant-utilisateur est le seul interlocuteur des utilisateurs car c'est à lui que parviennent toutes leurs commandes (CART: RES-5). C'est lui qui détient la liste de tous les utilisateurs potentiels du centre de calcul (JUREMA: RES-21). Tous les utilisateurs ayant des environnements personnalisés, il doit y avoir pour chacun d'eux:

- un identificateur pour le distinguer des autres;
- un mot de passe qui protège l'environnement;
- les ressources associées à l'utilisateur (fichiers, procédures, etc.);
- une priorité définissant ses prérogatives;
- etc.

Il est également nécessaire au correspondant service de connaître un minimum d'information concernant les applications; un aspect important de son rôle est en effet d'interpréter des requêtes d'accès à des applications. A son niveau, sont donc recensées à un instant donné toutes les applications disponibles dans le centre.

Ces deux types d'information reflètent bien le rôle de médiateur (aiguilleur) que joue le correspondant dans l'initialisation des dialogues des utilisateurs avec les applications.

1.3. Les moyens de communication

La communication entre le Correspondant-Service et un service, et entre le Correspondant-Utilisateur et un utilisateur, se fait à travers des VLC (Voie Logique de Communication). Une VLC est un ensemble qui permet à deux entités de même niveaux de communiquer en garantissant un taux d'erreur minimal et un contrôle de flux des informations qui transitent (TOSAN: RES-36). Elle se décompose en un ou plusieurs ensembles, chacun comprenant:

- le support de transmission: ligne, bus, ensemble modem ligne - modem;
- la gestion de transmission, c'est à dire gestion physique du support et gestion de la procédure de transmission;
- la gestion de protocoles de communication.

C'est par l'intermédiaire des VLC que les communications entre entités du CCR peuvent s'effectuer. Les informations qu'elles permettent d'échanger leur sont transparentes.

L'utilisateur dispose d'un langage de commande unique appelé LCE (Langage de Commande Externe) (TOSAN: RES-36). C'est dans ce langage que l'utilisateur s'adresse au Correspondant et c'est à travers lui qu'est vu le CCR. Il s'agit d'un langage:

- d'une forme simple et extensible: suites de requêtes encadrées par un début et une fin de session (commande := nom + liste de paramètres);
- interprétatif: toutes les commandes sont interprétées;
- interactif: il permet à l'utilisateur de dialoguer avec le correspondant.

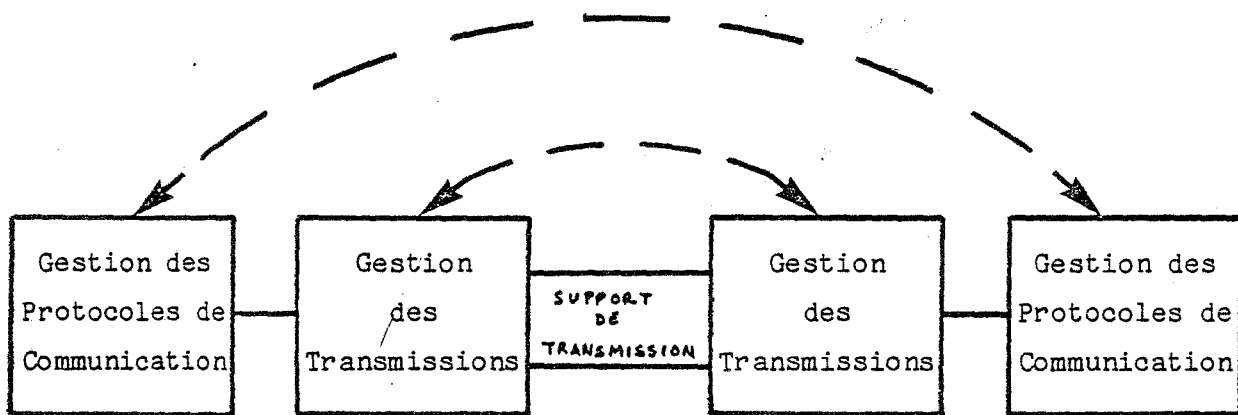


Figure - 2. Schéma général d'une Voie Logique de Communication (VLC).

La fonction principale du LCE est donc de masquer aux utilisateurs la variété des langages de contrôle existant sur les mini-ordinateurs du réseau et leur offrir une vision plus simple du CCR. Il doit permettre à l'utilisateur:

- d'entrer sous le CCR (LOGIN) et d'en sortir (LOGOUT);
- d'appeler les services, leur envoyer des données et en recevoir;
- d'utiliser des procédures cataloguées;
- d'échanger des messages avec d'autres utilisateurs;

A un moment donné, la commande devra parvenir au système d'exploitation supportant l'application sous la forme qu'il attend, c'est à dire dans son langage de contrôle. Ainsi a été conçu le LCI (Langage de Commande Interne), un intermédiaire entre le LCE et le langage de contrôle de chaque système d'exploitation (TOSAN: RES-36). De même, le LCI est utilisé pour le dialogue entre applications.

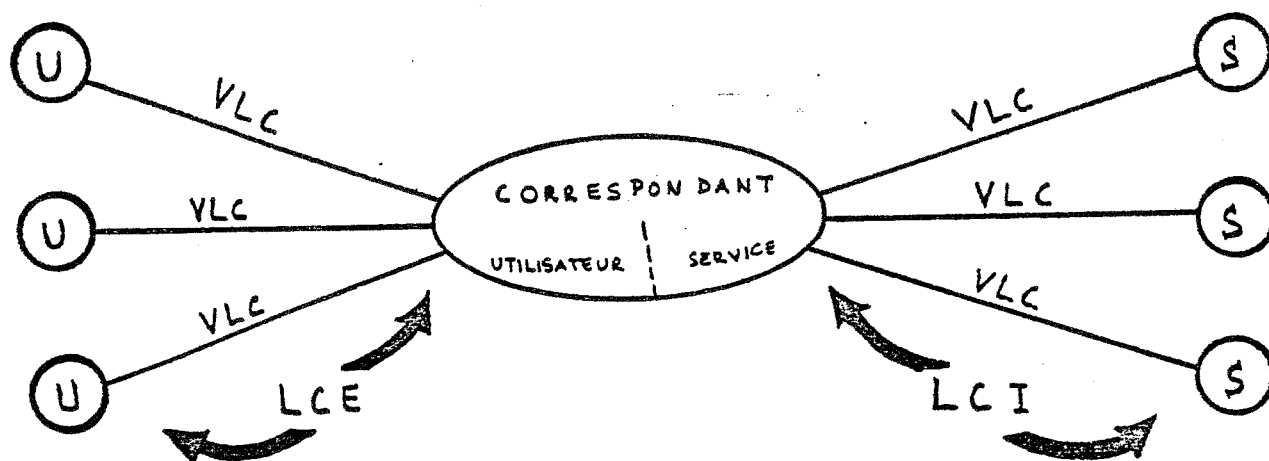


Figure - 3.

Afin de n'avoir qu'une version de l'interprète LCE, pour des raisons d'encombrement et de maintenance, celui-ci est localisé sur le Frontal (CART: RES-5). Quant à l'interprétation du LCI, elle est locale aux applications qui ont besoin de cet environnement pour dialoguer avec les autres applications.

1.4. Approche fonctionnelle

Le Centre de Calcul Réparti se compose de trois fonctions fondamentales (CHAMBON et al.: RES-7) :

1) La fonction "Frontal" : elle joue le rôle de concentrateur-diffuseur de terminaux locaux ou distants et celui de "plaque tournante" c'est à dire, de relais dans les dialogues entre les différents éléments du réseau. Elle regroupe essentiellement le correspondant et l'interprète du langage de commande.

2) Les services réseaux : ils sont en nombre quelconque. Un service réseau comprend trois niveaux :

Service local : c'est l'application offerte par la machine hôte (compilateur, gestion de fichiers locale, éditeur de texte, programme de gestion ou de calcul scientifique ...). Cette application est en général fournie par le constructeur ou écrite par un programmeur d'applications. L'utilisateur a l'impression de dialoguer avec ce niveau lorsqu'il est connecté à un service.

Environnement service local : ce sont les ressources nécessaires à l'exécution du service local. Ce niveau ne connaît que le langage de contrôle du système d'exploitation (O.S.) local;

Enveloppe réseau : c'est ce niveau qui assure une certaine ouverture de l'application locale sur l'extérieur. Il reçoit les commandes en LCI et effectue la correspondance avec le langage de contrôle local. C'est ce niveau qui connaît le correspondant service et dialogue avec lui.

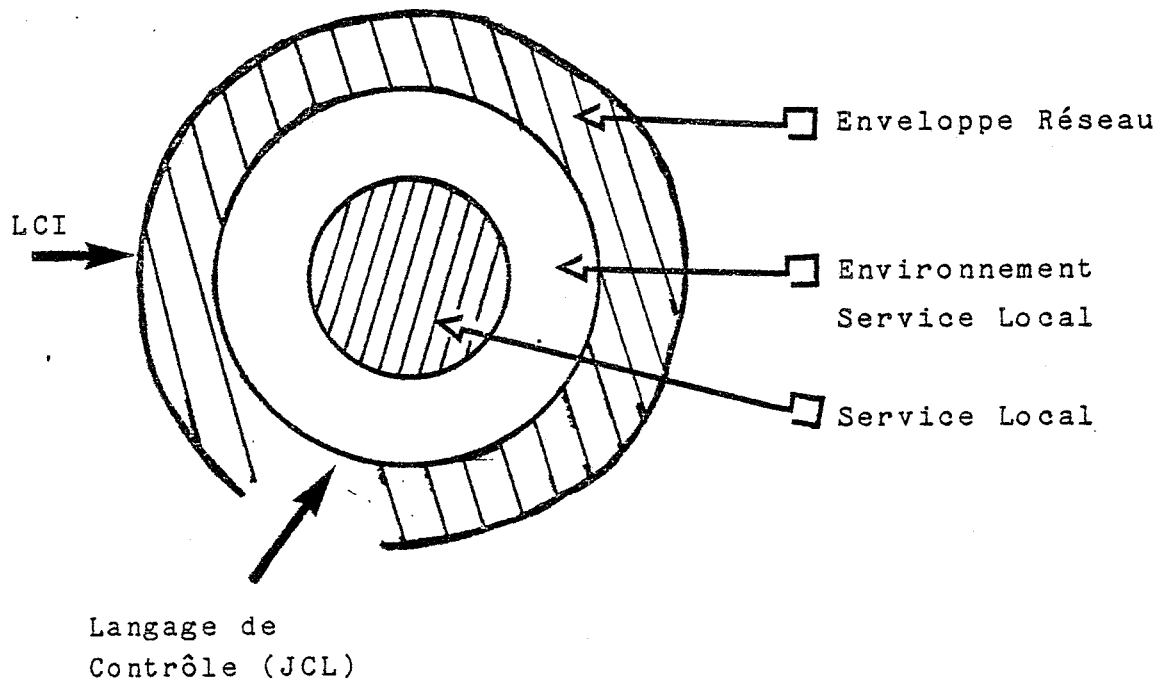


Figure - 4. Contexte d'un service réseau.

3) L'application fichier : c'est un service particulier dont la principale caractéristique est qu'il est réparti sur l'ensemble des ordinateurs connectés. C'est elle qui constitue l'objet principal et le moteur de cette étude.

L'utilisateur a une vision externe du CCR, qui ne l'oblige pas à connaître les particularités de chaque système d'exploitation pour pouvoir accéder aux services qu'ils offrent.

L'image d'un seul ensemble fonctionnel que le CCR veut donner à l'utilisateur est fondée sur la transparence de la localisation des services. A partir de son terminal, l'utilisateur peut activer des services situés sur n'importe quel ordinateur du CCR sans être contraint de préciser où ils se trouvent. Or, certains de ces services pouvant utiliser des ressources fichiers supportées par un autre mini, il semble évident que la localisation de ces fichiers devrait elle aussi être transparente à l'utilisateur. En effet, la transparence de localisation des services n'est pas très utile si elle n'est pas étendue aux ressources fichiers.

1.5. La fonction d'interprétation

La fonction d'interprétation qui donne à l'utilisateur l'impression d'être en face d'une machine unique est assurée à la fois par l'interprète et par les enveloppes réseaux des services. Cette interprétation comprend différents niveaux. Mais conceptuellement on distingue une même fonction d'interprétation dont les différents niveaux de réalisation peuvent prendre des formes diverses.

En effet certains niveaux sont soit distribués (cas des enveloppes localisées sur les sites des services) soit regroupés au sein d'une même entité : l'interprète.

En particulier le niveau d'interprétation qui fait la reconnaissance des ressources fichiers dialogue avec l'entité "application fichier". Il lui soumet des requêtes relatives aux fichiers. L'application fichier les exécute suivant un algorithme et des ressources qui lui sont propres. Elle transmet en retour un compte rendu au niveau d'interprétation concerné.

2. Interaction entre le CCR et le SCSGF

Le SCSGF a la même vocation que le CCR. Il est vu par celui-ci comme un outil capable de lui permettre de parachever sa fonction de "banalisation de ressources". A cette fin, deux places différentes, dans le CCR, ont été envisagées pour le situer, de façon à faciliter son interaction.

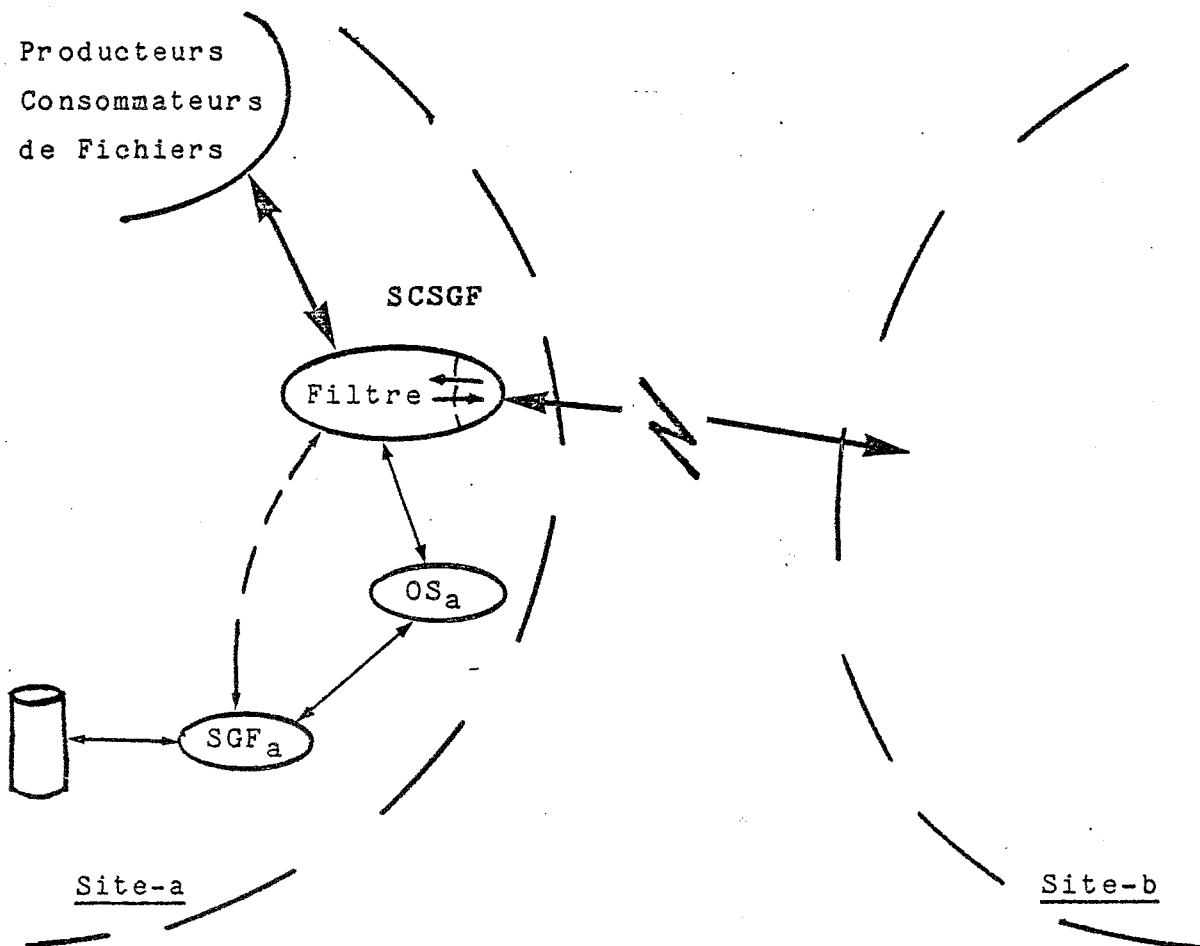


Figure - 5.

2.1. Localisation du SCSGF dans le CCR

Le correspondant-service étant la plaque tournante de toute communication entre ou vers les services, il semble évident que c'est là que devrait se situer le SCSGF. La récupération et l'analyse de toute commande concernant les ressources fichiers seraient ainsi facilement effectuée. Cette solution a aussi l'avantage de n'avoir qu'une seule version du SCSGF, ce qui a comme conséquence une simplicité de maintenance et de rajout ou retrait d'un nouveau système.

Une partie du correspondant-service étant déportée sur l'enveloppe réseau de chaque service, ceci a été pris en compte vis-à-vis d'une deuxième possibilité de localisation du SCSGF. En effet, étant donné qu'il n'existe qu'un seul service gestionnaire de fichiers (SGF) sur chaque site, l'idée a été de confier à son enveloppe réseau le filtrage des commandes qui lui parvenaient. Ses caractéristiques pouvant être assez diverses, ceci permettrait mieux leur prise en compte et libérerait le correspondant-service d'une tâche spécifique à quelques services seulement.

L'option a été prise en fonction, d'une part de la "décharge" du correspondant-service d'une tâche concernant peu d'éléments dans le CCR (un par site) et d'autre part, de la proximité du SCSGF, des services gestionnaires de fichiers. Nous avons donc choisi la deuxième option, qui offre aussi l'avantage de n'intercepter que les commandes relatives aux fichiers et pas toutes celles passant par le correspondant-service. Ainsi, le SCSGF faisant partie de l'enveloppe réseau du service SGF sera le mieux placé pour intercepter les commandes qui lui sont destinées.

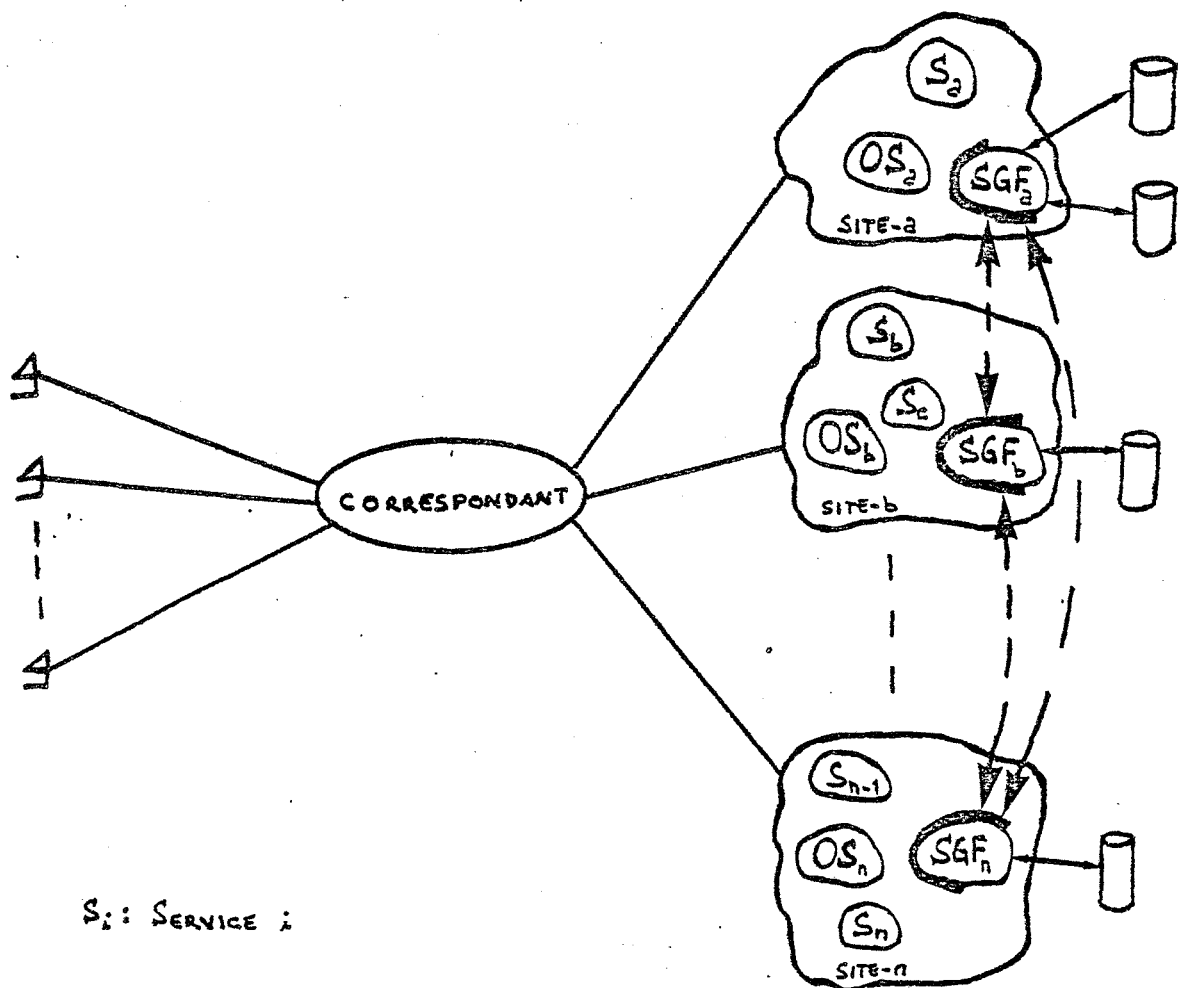


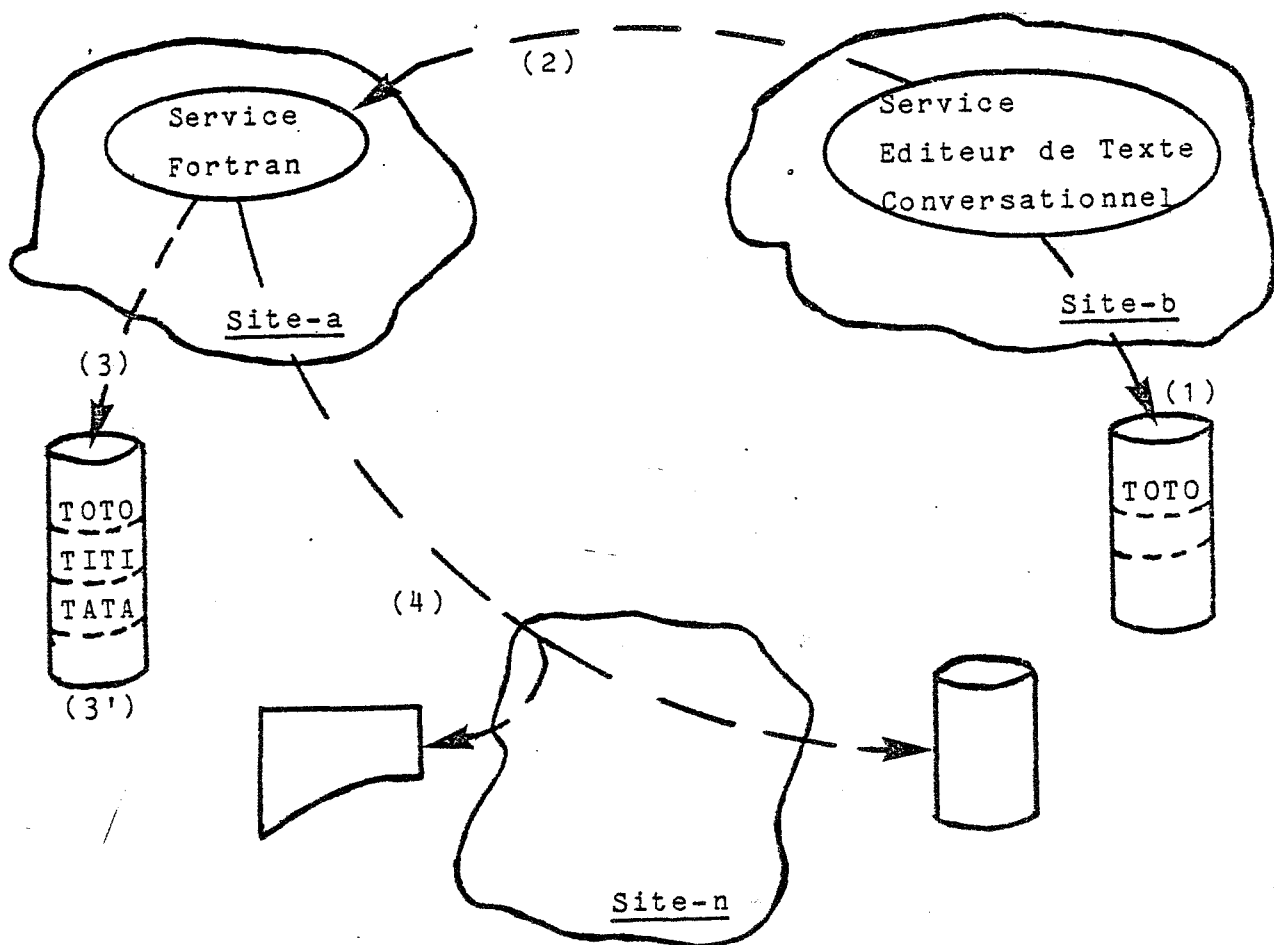
Figure - 6. Localisation du SCSGF dans le CCR.

2.2 Exemple d'utilisation possible

Dans un souci de clarté nous illustrerons la vision actuelle que l'utilisateur a du CCR (avec l'adjonction du SCSGF) par un exemple simple. L'extension de cet exemple à des cas particuliers reste toujours possible si elle prend en compte les considérations présentées jusqu'ici.

Soit un utilisateur du CCR en train de mettre au point des programmes en FORTRAN. A ce stade de son travail, cet utilisateur fait essentiellement de l'édition de texte, de la compilation et de l'impression de listes.

C'est ainsi que sans le savoir, l'utilisateur met en oeuvre des services qui peuvent ne pas être localisés sur la même machine. En effet l'éditeur de texte conversationnel (CONV) se trouve sur le site b. Ce service produit en sortie un fichier source TOTO.FOR de type FORTRAN; ce fichier est stocké sur le même site que le service qui l'a créé (site b). Après cette phase d'édition de texte, l'utilisateur se connecte au service FORTRAN qui se trouve sur le site a.



- (1) Modification du fichier source
- (2) Transfert du fichier source en entrée du service FORTRAN
- (3) Exécution du service FORTRAN en local
- (3') Production de fichiers (permanents et temporaires)
- (4) Transferts des fichiers temporaires en sortie

Figure - 7. Exemple d'utilisation.

Il tape alors la commande:

```
COMPILER  FICHIER SOURCE  = TOTO.FOR
          FICHIER OBJET   = TITI.OBJ
          FICHIER LISTING = TATA.LST
```

Il s'attend alors aux actions suivantes:

- compilation du fichier source TOTO.FOR,
- rangement du code objet généré dans le fichier TITI.OBJ;
- production d'un fichier TATA.LST qui contiendra la liste source et le diagnostic d'erreurs.

L'enveloppe réseau du compilateur FORTRAN doit analyser les paramètres de la commande, en extraire les noms des fichiers. Ensuite elle s'adresse au Système de Coopération de SGF. Ce dernier s'assure que le fichier source est présent sur le site du compilateur; sinon, il demande son transfert (JUREMA, NIANG: RES-22). Pour les nouveaux fichiers à créer, le Système de Coopération doit s'assurer qu'il n'y aura pas d'ambiguïté: existence sous le même numéro de compte d'un fichier de même nom réseau. Il est bien entendu que ce nom réseau universel identifie entièrement et sans ambiguïté tout fichier créé et accessible du CCR. Après ces vérifications et un certain nombre d'autres vérifications (protections, droit d'accès, disponibilité de l'espace de stockage temporaire etc.) on attend la fin du transfert des fichiers en entrée.

Dès lors, toutes les conditions sont réunies pour lancer le service de compilation. Ce dernier évolue dans son environnement local.

A la fin de l'exécution du service en local, il peut s'avérer nécessaire de faire encore appel au Système de Coopération pour transférer les fichiers en sortie qui sont destinés à être stockés sur d'autres sites. Ce transfert concerne les fichiers dont le site de stockage est différent de celui de l'application. Ces fichiers avaient été transférés sur ce site pour y subir une simple modification.

Au cours de la session de travail décrite précédemment, le comportement type de l'utilisateur a consisté en:

- une phase de modification d'un programme source. Cette phase faisait appel au service éditeur de texte;
- une phase de compilation du programme source faisant appel au service FORTRAN;
- l'impression de la liste source par la ressource imprimante.

Les différentes phases de cette session de travail font donc appel à des services et des ressources qui peuvent être localisés sur des ordinateurs différents.

Cependant, il faut faire en sorte que pour l'utilisateur tout se passe comme s'il travaillait sur une seule machine. Cette machine virtuelle doit avoir des possibilités et des performances comparables à celles de chacun des mini-ordinateurs pris individuellement dans les domaines où il excelle.

Aussi, vouloir garder la transparence de la localisation des services et des ressources suppose la mise en place de plusieurs mécanismes internes au CCR. Ces mécanismes inconnus de l'utilisateur doivent assurer cette transparence.

3. Extension à d'autres systèmes

Définir un Système de Coopération général, capable de s'intégrer à différents systèmes informatiques, exige un certain recul par rapport au problème posé. Autrement dit, pour qu'un tel système puisse satisfaire les besoins et contraintes imposés par plusieurs environnements-systèmes ayant des caractéristiques diverses, il doit avoir une vision assez globale de ces environnements. Cela étant, la généralité du Système de Coopération présenté n'induit pas son extensibilité à toute "famille" de systèmes.

Le SCSGF est, en effet, un outil orienté réseau. Son intérêt et applicabilité est d'autant plus grand que l'environnement qui le supporte est spécialisé, voire homogène. Ceci lui confère, à priori, une plus grande aptitude à satisfaire les réseaux locaux plutôt que les réseaux généraux.

C'est la compatibilité entre les différents ordinateurs participant au réseau qui déterminera le niveau du service rendu par le SCSGF. Par conséquent, le SCSGF est extensible particulièrement à tout système appartenant à la "famille" du CCR, c'est à dire du type réseau local, ou réseau général si les composants de celui-ci ne sont pas d'origines trop diverses.

CHAPITRE IX

DEFINITION DE L'EXTENSION DU CONTEXTE
D'APPLICATION DE L'OUTIL SCSGF

1. Motivations des réseaux d'ordinateurs
2. Contexte : les réseaux d'ordinateurs
 - 2.1. Réseau de communication
 - 2.2. Système réseau
3. Présentation de quelques exemples
 - 3.1. Réseau local
 - 3.2. Réseau général
4. Intérêt et applicabilité d'un SCSGF
5. Limitations

Le CCR (Centre de Calcul Réparti) est à l'origine de notre travail car c'est lui qui nous a sensibilisé aux problèmes de compatibilité et de coopération entre systèmes hétérogènes. L'étude que nous avons été amenés à faire sur ces problèmes (JUREMA, NIANG: RES-22) nous a conduits à élargir le cadre de notre travail, en fonction de son applicabilité à d'autres systèmes existants.

Ce travail, comme un certain nombre d'autres du genre, vient en réponse au besoin créé par l'interconnexion d'ordinateurs. De ce fait, un aperçu des origines et motivations des réseaux d'ordinateurs est donné afin de présenter son cadre. Puis, après un résumé de quelques exemples de réseaux, nous présentons l'apport du "Système de Coopération de Systèmes de Gestion de Fichiers", son intérêt et applicabilité et enfin, ses limitations actuelles.

1. Motivations des réseaux d'ordinateurs

L'évolution suivie par les systèmes informatiques témoigne d'un souci constant de minimiser le temps nécessaire à l'exécution, complète, de toute action déroulée sur un ordinateur. Plusieurs techniques ont été utilisées à cette fin (traitements par lots, temps partagé, etc.) et un progrès relatif a été fait. Malgré cette réussite initiale un nouveau seuil, difficile à franchir, est apparu. Le simple partage de temps du processeur entre les utilisateurs ne suffisait plus. Il fallait désormais des techniques plus évoluées.

Toute action (programme utilisateur ou programme d'application, etc.) a généralement besoin, pour son déroulement normal, d'un certain nombre de ressources. Ces ressources (imprimante, fichier, article d'un fichier, etc.) étant utilisées en série, l'une après l'autre, le nouvel objectif fixé a été de réduire le temps d'attente de traitement de chacun d'eux. De plus, pendant que la ressource est dédiée à une action précise, tout sur l'ordinateur reste disponible, en attendant une éventuelle sollicitation.

L'idée de la nouvelle technique a été d'accroître le partage de ressources et de réaliser certaines actions en "parallèle". Nous avons donc vu apparaître les systèmes multi-processeurs, les ordinateurs en "tandem" ou partageant de la mémoire, etc. Cela était peut-être le prélude des réseaux d'ordinateurs actuels.

2. Contexte: les réseaux d'ordinateurs

Le terme "réseau d'ordinateurs" évoque l'interconnexion de systèmes informatiques, isolés et indépendants, et la mise à leur disposition d'un moyen de communication dont la gestion est prise en charge, complètement, par le réseau (ELOVITZ, HEITMEYER: RES-15). Quant aux ressources offertes par chacun des ordinateurs, elles peuvent être gérées soit par les ordinateurs qui les abritent, soit par le réseau.

2.1. Réseau de communication

Lorsque la gestion des ressources du réseau est réalisée par chacun des ordinateurs qui les supporte, la vision que l'utilisateur a du réseau est celle d'une

collection d'ordinateurs ou systèmes, offrant des services et des ressources variés accessibles à partir de différents points du réseau. L'utilisateur doit cependant connaître les langages de commande de tous les systèmes auxquels il veut accéder, ainsi que le schéma de localisation des ressources sur les différents ordinateurs. En effet, les procédures de connexion logique aux différentes applications et leur mode d'utilisation sont dépendants des systèmes hôtes. Etant donnée la limitation de ce type de réseau à un "service de communication", il ne constitue qu'un réseau de transport qui correspond aux niveaux 1, 2, 3 et 4 du modèle ISO : nous l'appellerons Réseau de Communication.

2.2. Système réseau

Un Système Réseau ne prend pas seulement à sa charge l'aspect communication mais joue également un rôle d'intermédiaire plus actif entre les utilisateurs et les applications (CART: RES-5). Il gère les ressources offertes par chacun des ordinateurs participant au réseau offrant ainsi à l'utilisateur une transparence complète de la localisation des ressources. Ceci implique évidemment la mise en oeuvre d'un langage de commande unique, compréhensible par chacun des ordinateurs connectés (TOSAN: RES-36). Ce "Langage de Commande Réseau" permet à l'utilisateur d'accéder au réseau en se connectant au Système Réseau plutôt qu'à un système particulier du réseau.

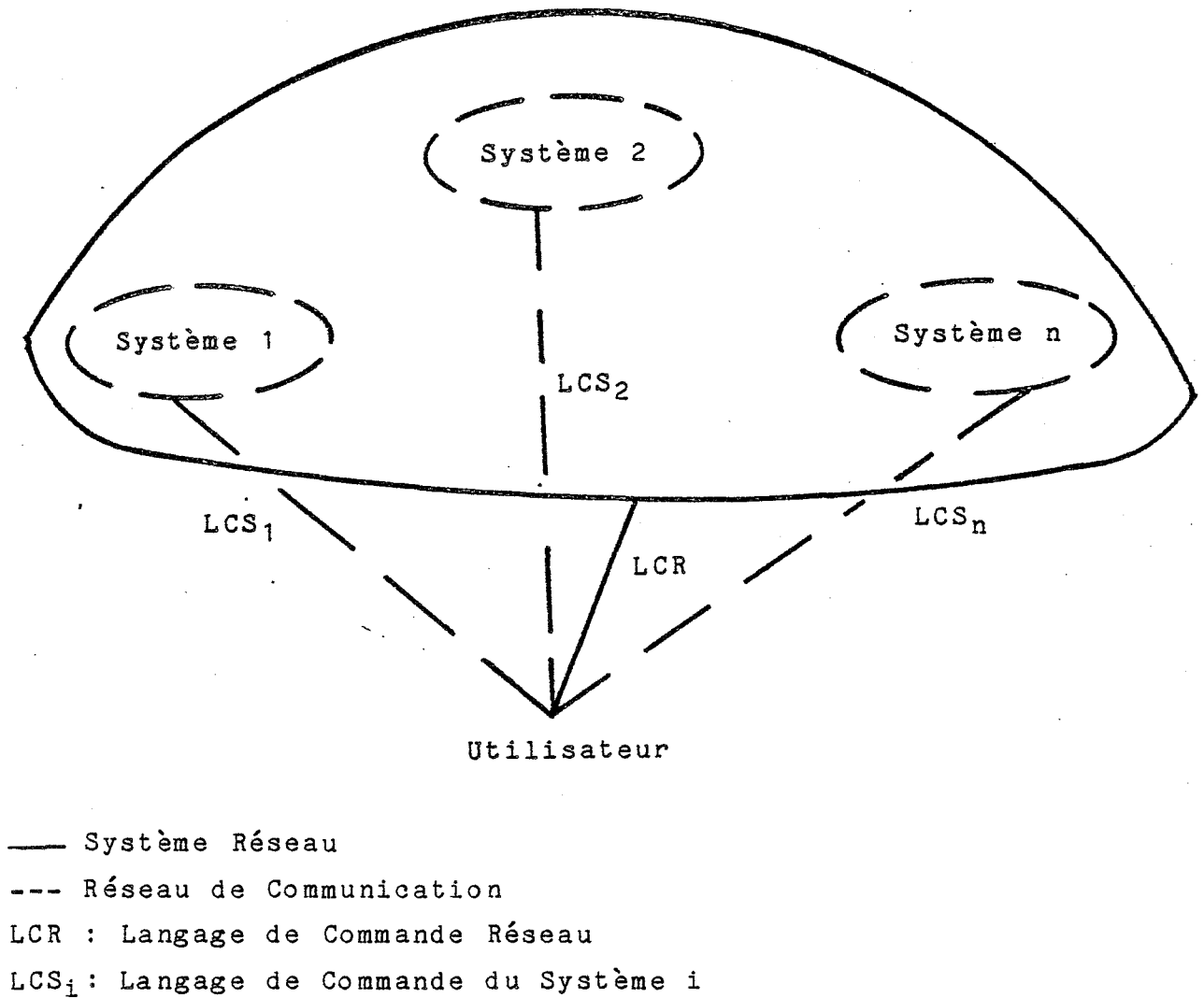


Figure - 1. Réseau d'ordinateurs.

3. Présentation de quelques exemples

A la notion de degré de transparence offert à ses utilisateurs, soit par un système réseau ou par un réseau de communication, s'ajoute la notion de "Réseau Local" et "Réseau Général". Ces deux types de réseaux informatiques appartiennent, souvent, aux classes système réseau et réseau de communication, respectivement.

La conception de certains Réseaux Locaux à des fins spécifiques est la raison pour laquelle on les trouve, de temps à autre, avec la dénomination de "réseau local spécialisé". Ils se caractérisent surtout par une certaine unité géographique, c'est à dire une distance faible entre les différents composants, et un nombre restreint d'ordinateurs participant au réseau (WULF, LEVIN: RES-38) (ELOVITZ, HEITMEYER: RES-15).

Par opposition aux "Réseaux Locaux spécialisés", les Réseaux Généraux permettent l'accès à tout type d'application et leurs composants sont souvent dispersés sur un espace géographique très étendu. Les réseaux généraux se caractérisent par l'aspect hétérogène de leurs composants d'une part et d'autre part, par la connaissance que l'utilisateur doit avoir de la localisation des ressources auxquelles il veut accéder. L'utilisateur doit en plus connaître le Langage de Commande du site qui supporte ces ressources.

Les principaux buts des réseaux informatiques sont:

- partage des ressources: il permet à un plus grand nombre d'utilisateurs d'accéder à un plus grand nombre de ressources;
- flexibilité: il rend aisé l'insertion ou le retrait d'un centre d'activité quelconque du réseau;
- fiabilité: il assure que le mauvais fonctionnement de l'un des sites ne perturbe pas le réseau tout entier.

La compatibilité des composants d'un réseau homogène n'étant pas suffisante pour les faire coopérer, à plus fortes raisons un système de coopération intéresserait

aussi les réseaux hétérogènes. Nous donnons donc par la suite quelques exemples de Réseaux Locaux et Réseaux Généraux, indépendamment de ces deux aspects.

3.1 Réseau local

Les Réseaux Locaux existant actuellement, opérationnels ou en phase de réalisation, ont des caractéristiques diverses. Ceci rend difficile toute tentative de les analyser de manière précise. Voici quelques exemples de réseaux locaux:

a) MININET (MANNING, PEEBLES: RES-25): ce réseau homogène a la structure d'une boucle à laquelle sont connectés plusieurs mini-ordinateurs. Chaque site participant est relié à la boucle par un noeud et tous les messages qui circulent sont examinés pour vérifier si l'adresse de leur destinataire correspond ou non à celle d'un site participant. Chaque mini contrôle un certain nombre de terminaux qui peuvent accéder à toutes les données distribuées sur le réseau. Ce réseau est orienté exclusivement vers le traitement de transactions dans une Base de Données physiquement répartie.

b) DCS (FARBER et al.: RES-17) (FARBER: RES-16): (Distributed Computer System) il s'agit d'un système réseau de mini-ordinateurs hétérogènes reliés entre eux par un anneau de communication. Un processeur matériel appelé "Ring Network" (RI), réalise l'interface entre un mini-ordinateur et l'anneau. Sur ce réseau aucun site n'est privilégié; les messages sont déposés sur l'anneau et l'interface "RI" associée à un site vérifie si un message qui vient d'arriver

est destiné à un processus de son site; si oui, il le remet à son site après l'avoir recopié et acquitté vers le site émetteur où il sera détruit; si non, il le transmet à l'interface suivante sur l'anneau;

c) M2 (HEBENSTREIT: RES-20): (Multi-Mini-Ordinateurs) dans ce projet de système réseau les différentes fonctions du logiciel sont confiées à des mini-ordinateurs hétérogènes spécialisés. On distingue ainsi:

- les "mini exécutants",
- le "mini maître" auquel sont connectés tous les périphériques d'entrée-sortie et qui supervise le fonctionnement global du système,
- le "mini fichier".

Tout programme utilisateur écrit en langage évolué fait l'objet d'une compilation pour laquelle le langage objet est un langage commun LOC (Langage Objet Commun). Il existe sur chaque mini exécutant un transposeur c'est à dire, un traducteur de langage LOC en langage machine. Ceci doit permettre à un travail utilisateur (décrit dans un langage de scénario) d'être exécuté sur n'importe quel mini.

c) ARAMIS (BEAUFILS et al.: RES-3);

f) HMINET (POPESCU-ZELETIN: RES-30);

g) ETHERNET (METCALFE, BOGGS: RES-27);

3.2. Réseau général

A partir du raccordement direct puis indirect, des centres d'activité qui souhaitent s'échanger des données, la plupart des réseaux généraux utilisent actuellement le principe de la commutation de messages. Un ordinateur M qui désire émettre un message pour un ordinateur N le transmet à l'un de ses voisins et le processus est itéré jusqu'au destinataire. Cette technique a encore évolué et la commutation réalisée sur les réseaux plus récents concerne des "paquets" d'information (MACCHI, GUILBERT: RES-24). Ces paquets ont une taille fixe et lorsque celle-ci est inférieure à la taille du message à envoyer, l'information doit être fragmentée avant l'envoi et reconstituée à la réception. Le nombre et la disposition géographique des commutateurs peuvent être établis selon diverses architectures. Celles-ci visent à "recouvrir" un certain espace géographique à un coût minimum.

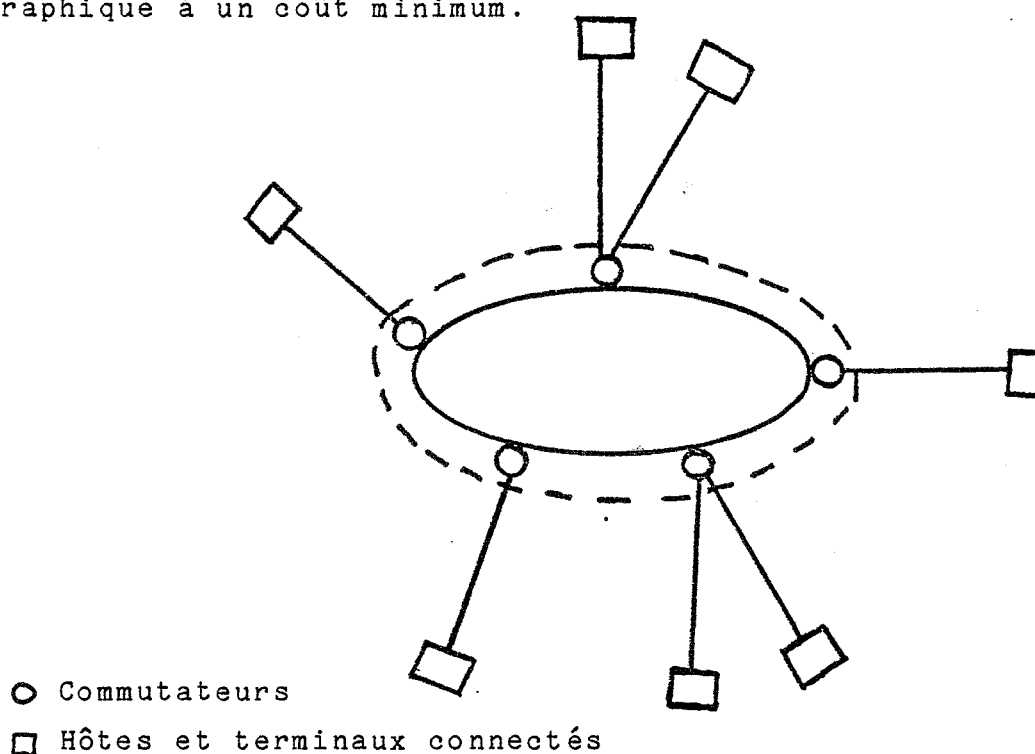


Figure - 2. Réseau de communication.

Les Réseaux Généraux ont une structure à deux niveaux:

- le réseau de transport ou réseau de communication qui offre aux usagers un service de transmission;
- les hôtes, qui sont les ordinateurs et terminaux connectés. Ils accèdent au système de communication et l'utilisent comme une boîte noire pour y déposer et retirer des messages.

La mise en oeuvre de tels réseaux nécessite des moyens considérables, ce qui explique, en partie, une participation généralement à un niveau national. Ci-dessous nous présentons quelques exemples de réseaux généraux:

a) ARPANET (ROBERTS, WESSLER: RES-23): Démarré en 1968 sous l'impulsion de "l'Advanced Research Project Agency", c'est un réseau reliant les Universités Américaines ayant des contrats de recherche avec l'ARPA. Premier réseau à deux niveaux, opérationnel dès 1972, utilisant la technique de commutation de paquets. Il s'est inspiré du réseau SITA (Société Internationale de Télécommunications Aéronautiques) et a permis un certain nombre d'expériences sur les types de routage.

b) CYCLADES (POUZIN: RES-31) (ZIMMERMANN: RES-39) : Projet Pilote Français commencé en 1972; il concerne la mise en oeuvre d'un réseau d'ordinateurs hétérogènes reliant universités et administration, orienté vers l'accès aux bases de données. On distingue nettement les deux niveaux:

- CIGALE : réseau de commutation de paquets géré par des mini-ordinateurs spécialisés (MITRA 15);

- les "hôtes" : ordinateurs plus importants ou concentrateurs de terminaux, raccordés à CIGALE par l'intermédiaire de "stations de transport" permettant le découpage, réassemblage des messages et leur acheminement vers les abonnés des sites qu'elles gèrent.

c) EIN (SEDILLOT: RES-35) : (European Informatic Network)

Réseau Européen construit à partir des enseignements apportés par CYCLADES qui a permis d'affiner les protocoles de transport et de lancer des propositions de standards au niveau des "protocoles de bout en bout".

d) TRANSPAC (TRANSPAC: RES-37) : Réseau national de transmis-

sion de données par paquets. Il a bénéficié des expériences des deux réseaux précédents. Globalement, il apporte un service de transport de paquets. C'est aussi un pas important vers la mise en oeuvre de normes. L'accent a été mis sur la fiabilité des noeuds (reconfiguration dynamique).

e) SNA (MCFADYEN: RES-26) (CORR, NEAL: RES-12) : (Systems

Network Architecture) l'architecture du réseau d'IBM définit un ensemble de concepts et protocoles en vue de l'interfonctionnement d'une large gamme de ses produits, matériels et logiciels.

f) DNA (CONANT, WECKER: RES-11) : (Digital Network Architec-
ture ou DECNET).

4. Intérêt et applicabilité d'un SCSGF

Nous venons d'avoir un aperçu des réseaux informatiques qui existent actuellement. En ce qui concerne les Systèmes Réseaux c'est à dire, ceux pour qui la localisation des ressources est transparente à l'utilisateur, ils sont peu nombreux. De plus, les services, répartis sur plusieurs sites, dont les différents composants coopèrent entre eux afin de pouvoir s'exécuter sont relativement rares et souvent encore en phase expérimentale (GIFFORD: GFR-5). Cela est, en partie, dû au fait que l'origine des matériels mis en relation n'est généralement pas connue en avance, ce qui rend difficile la prise en compte des particularismes de chacun.

Jusqu'à il y a peu de temps, les seuls Systèmes Répartis étaient soit des "systèmes constructeurs", fondés sur l'interconnection de matériels homogènes, soit des "expériences parcellaires", faites en tenant compte des particularités des matériels mis en jeu. Dans ce cadre les difficultés rencontrées lors d'une éventuelle adjonction d'un matériel hétérogène n'ont pas permis l'extension de tels systèmes.

Quant aux réseaux informatiques, quels qu'ils soient, ils ont tous, à un certain niveau, des entités qui ont besoin de coopérer entre elles. Or, le nombre de réseaux actuels qui permettent cette coopération, est encore restreint. De plus, les réseaux d'ordinateurs ayant des entités qui coopèrent sont souvent très spécifiques et dépendants de l'environnement. Il s'agit en général de quelques expériences ponctuelles de la part de certains utilisateurs ou alors d'outils fournis par certains constructeurs pour leurs propres systèmes.

Les réseaux actuels constituent, pour la plupart, un moyen de transport d'information entre ordinateurs. Ils offrent, à cet égard, des protocoles de communication et de transfert de données. Par ce biais les utilisateurs peuvent dialoguer entre eux et accéder à un plus grand nombre de ressources, à partir de n'importe quel point d'accès du réseau. Toutefois ceci implique des connaissances particulières de leur part car les applications, et leurs entités respectives, situées sur les différents ordinateurs connectés au réseau, ne coopèrent pas entre elles.

Les utilisateurs des Réseaux Généraux, par exemple, ne disposant souvent que d'un "service de communication", doivent connaître la localisation des ressources auxquelles ils veulent accéder et le langage de commande du site supportant ces ressources. En fait, les différents services disponibles sur le réseau ne se connaissant pas, il ne peut pas y avoir de coopération entre eux. Par conséquent, lorsque l'utilisateur a besoin de deux ressources situées sur des sites distincts et qui doivent être utilisées par son application sur un troisième site, c'est à lui d'envoyer une demande de ressources aux sites concernés et d'activer son application, une fois que les ressources dont elle a besoin seront sur place. Cela correspond généralement à un volume d'information important que l'utilisateur doit appréhender et un certain nombre de manipulations qu'il doit mettre en jeu. Ce problème prend des proportions encore plus importantes, dès lors que deux (ou plus) réseaux informatiques s'interconnectent. L'attention donnée à ce problème, soit par des études de techniques d'interconnexion (GIEN, ZIMMERMANN: MAS-10), (ANDERSON, JENSEN: MAS-1), soit par la définition de normes internationales (RYBCZYNSKI, PALFRAMAN: RES-34), (GROSSMAN, al.: RES-19), fait espérer que les prochains réseaux informatiques auront moins de difficultés à offrir des "Systèmes Répartis" à leurs utilisateurs.

L'avenir des Systèmes Répartis s'avère donc prometteur compte tenu des prochains Réseaux Informatiques qui seront développés. En effet, les Réseaux Informatiques actuels sont, presque tous, constitués de systèmes centralisés qui communiquent entre eux mais ne sont pas conçus pour coopérer.

En attendant la venue des systèmes fédérables et coopératifs, le développement d'un outil capable de faire coopérer des services ou systèmes déjà existants nous semble, à priori, la solution à adopter. Evidemment les logiciels des constructeurs doivent être touchés au minimum afin que leur pérennité soit conservée et que l'utilisateur puisse ainsi profiter des mises à jours réalisées par le constructeur. De tels systèmes de coopération rendent la localisation des ressources du réseau transparente à l'utilisateur et leur développement est d'autant plus aisé, qu'il y a plus d'équipements homogènes dans le réseau.

En ce qui concerne les ressources fichiers, une solution possible à ce problème que nous venons d'évoquer est l'utilisation d'un "Système de Coopération de Systèmes de Gestion de Fichiers". Ceci constitue un outil essentiel aux réseaux informatiques. L'importance et l'applicabilité d'un SCSGF ne varie, d'un type de réseau à un autre, que par le nombre des bénéficiaires (celui-ci est normalement plus élevé pour les Réseaux Généraux par exemple mais nécessite en revanche un investissement plus important).

5. Limitations

Selon les caractéristiques de certains réseaux, indépendamment de leur type, l'apport d'un Système de Coopération de Systèmes de Gestion de Fichiers (SCSGF) est

d'une importance plus ou moins grande. Il ne s'agit pas d'un outil "ad hoc" mais plutôt d'une solution générale et à court terme, pour résoudre un problème parfois assez important.

L'intérêt d'un SCSGF n'étant pas systématique, il doit être étudié dans chaque cas de figure. En effet, étant donnée l'évolution rapide des systèmes informatiques actuels et particulièrement des Bases de Données Réparties, l'intérêt est d'autant plus grand que le temps "d'implémentation" est petit. Le SCSGF étant un palliatif, l'effort et le temps nécessaire à sa mise en exploitation ne doivent pas être très importants. Ainsi, les considérations les plus importantes pour son application à un système donné, sont sur les aspects suivants:

- origine des matériels : de ceci dépend la variété des types de fichiers pour lesquels le SCSGF contribuera à la banalisation. Dans un système hétérogène le SCSGF sera limité à certains types de fichiers séquentiels tels que les fichiers "source". Dans un contexte homogène, un tel dispositif permettrait d'étendre la banalisation à l'ensemble des fichiers, quels qu'ils soient.

- délais de transmission : ceci tend à limiter l'application du SCSGF aux Réseaux Locaux. Le temps de transfert d'un fichier d'un site à un autre et le dialogue nécessaire à cela font que le SCSGF est considéré plus orienté Réseau Local que Réseau Général, à moins que celui-ci ait des caractéristiques un peu particulières (homogène, lignes spécialisées, extension géographique relativement restreinte, etc.). Il faut également prendre en compte les nouvelles technologies de transmission de données à haute vitesse.

CONCLUSION

1. Bilan de l'étude

- 1.1. Principaux centres d'intérêt
- 1.2. Apport d'un système de coopération

2. Prospectives

- 2.1. Bases de données et Fichiers
- 2.2. Architectures des systèmes répartis
- 2.3. Aspect méthodologie
- 2.4. Conditions d'intégration de la coopération

Notre ambition initiale était de traiter la question de coopération entre services distants, plus précisément de faire coopérer plusieurs Systèmes de Gestion de Fichiers (SGF) situés sur des sites distincts appartenant à un même réseau local, le CCR (Centre de Calcul Réparti).

Ne pouvant pas passer sous silence certains aspects de l'Informatique Répartie nous avons aussi porté notre réflexion sur certaines de ses implications les plus importantes. Ainsi, les conclusions sur cette étude seront divisées en deux parties. D'abord nous faisons un bilan de l'étude avec la présentation des principaux centres d'intérêt. Puis viennent les prospectives.

1. Bilan de l'étude

Les Systèmes Répartis recouvrent une gamme de sujets très grande et variée. Les rapports entre certains de ces sujets peuvent sembler difficiles à établir, selon qu'on met en évidence tel ou tel autre de ses aspects. Le tout constitue un volume d'information difficile à appréhender, d'autant plus que les frontières entre les différents domaines ne sont pas bien définies. Par exemple, il n'est pas rare de voir quelqu'un qui travaille sur les Bases de Données Réparties être mêlé à des problèmes de Réseaux.

1.1. Principaux centres d'intérêt

Afin de mieux préciser les implications d'un Système Réparti, plusieurs de ses aspects ont été pris en compte. Quelques exemples ont été utilisés pour illustrer certains de ces aspects. Un cas particulier, celui du SCSGF (Système de Coopération de Systèmes de Gestion de Fichiers) du CCR, a servi comme exemple de référence tout au long de cet ouvrage. La démarche, selon laquelle nous avons procédé, a consisté, dans les grandes lignes, à traiter les points suivants:

a) Cadre de l'étude: un bref historique des Systèmes Répartis a été présenté. Les réseaux d'ordinateurs étant à son origine, un aperçu de leur évolution est donné. Le CCR a été utilisé comme exemple de Réseau Local hétérogène. Dans un tel système, la question de répartition ou coopération, se pose. En ce qui concerne des services similaires tels les différents SGF existant dans le réseau par exemple, faut-il les ignorer et en conserver un seul réparti, ou simplement faire coopérer ceux qui existent déjà? Ceci étant l'objet principal de notre étude, une présentation détaillée des SGF en général, leurs points communs et incompatibilités, a été faite.

b) Les Systèmes de Gestion de Fichiers et Bases de Données répartis: en ce qui concerne les Fichiers Répartis, après la présentation de quelques exemples importants, nous avons fait une étude comparative; quant aux Bases de Données Réparties nous avons procédé de façon inverse: nous avons d'abord présenté les deux principales approches de la répartition et nous les avons illustrées par un panorama des principaux systèmes existants.

c) Mécanismes d'exécution répartie et le SCSGF: la communication entre deux entités quelconques étant étroitement liée à l'utilisation d'un protocole donné, nous avons donc décidé de l'étudier en détail. Ensuite nous avons présenté le SCSGF, les entités en présence et les types de dialogue possible entre elles, leur localisation, la caractéristique et l'apport du SCSGF.

d) Modèles d'architecture: nous y avons présenté quelques modèles d'architecture, la nécessité d'un modèle commun, un modèle de référence qui constitue déjà un effort de normalisation, et nous terminons par la présentation de l'architecture du SCSGF et ses entités internes.

e) Applicabilité d'un SCSGF: nous présentons son intérêt et applicabilité pour le CCR d'abord et ensuite son extensibilité à d'autres systèmes. Etant donnée son applicabilité potentielle dans les réseaux informatiques, nous présentons pour conclure, les différents types de réseaux existants et le degré d'intérêt qu'un SCSGF peut avoir pour chacun d'eux.

1.2. Apport d'un système de coopération

Nous ne prétendons pas, avec cette étude, avoir résolu tous les problèmes inhérents aux Systèmes Répartis mais au moins avoir éclairci un certain nombre et résolu quelques uns. Les besoins déjà pressentis que le développement rapide des réseaux et bases de données accroissent considérablement, nécessitent qu'on s'y intéresse de plus en plus.

La facilité de coopération entre plusieurs services distants reste fonction de l'origine des équipements mis en relation, (même constructeur ou pas), et de son degré d'ouverture vers l'extérieur. Nous apportons simplement, en attendant une normalisation plus vaste et précise, une étude générale et une méthode particulière afin de faire coopérer des services distants, plutôt que de les refaire, de toutes pièces, répartis.

L'ampleur du thème, Système Réparti, nous a inéluctablement conduit à consulter une importante bibliographie dont nous nous sommes efforcés d'extraire l'essentiel sans perdre de vue notre préoccupation majeure. Cependant celle-ci ne nous empêche pas d'avoir une vision d'ensemble des systèmes répartis en général.

2. Prospectives

2.1. Bases de Données et Fichiers

En observant les efforts qui sont faits pour faire coopérer Systèmes de Gestion de Bases de Données et Systèmes de Gestion de Fichiers, on est davantage persuadé de l'actualité de ce phénomène, même si les premiers ont parfois tendance à éclipser les seconds.

Pourtant les Systèmes de Gestion de Bases de Données reposent toujours sur des Systèmes de Gestion de Fichiers. Assez souvent la coopération entre SGBD est assurée au niveau des Systèmes de Gestion de Fichiers (DISCO, FRERES) car vu l'hétérogénéité des SGBD participants c'est le seul niveau susceptible de présenter moins de disparités.

Une base de données peut être caractérisée comme étant un ensemble de données ayant des liens sémantiques entre elles et représentées d'une certaine façon au niveau interne et pas nécessairement vues de la même manière au niveau externe. Un SGBD peut être défini comme un système assurant la transparence de la représentation physique des données et permettant leur description, leur stockage et leur manipulation en garantissant l'indépendance entre données et programmes. Les Systèmes de Gestion de Bases de Données Répartis ajoutent à ces fonctions, celle de la gestion de la transparence de la localisation.

Sur ce point, l'avenir appartient certainement aux Machines Bases de Données (GARDARIN et al. MAS-8) en environnement réparti. Celles-ci associent les progrès faits en matière de technologie des composants (VLSI, mémoires associatives, architectures multiprocesseurs incorporant des noyaux de systèmes, mémoires à bulles, disques optiques...) à ceux en cours en matière de techniques de gestion de Bases de Données (modèles de données, langages de manipulation, contrôle de cohérence de l'information ...).

2.2. Architecture des systèmes répartis

Il est fréquent de constater dans les architectures de systèmes répartis, une absence d'uniformité dans le découpage entre les différents sous-systèmes :

- Sous-système de traitement
- Sous-système de stockage

- Sous-système de communication

- Sous-système d'accès.

Ceci se reflète au niveau de l'architecture du système global et rend toute tentative de coopération très ardue. Cette absence de coordination dans la façon d'appréhender le problème se répercute au niveau pratique, car si certains problèmes pratiques qui se posent sont résolus à un certain niveau par un constructeur, ils seront résolus à d'autres niveaux par ses homologues. Ainsi il ne suffit pas de changer le niveau en question.

Bien qu'un regroupement de certaines fonctions soit parfois rendu nécessaire pour des raisons de performances, au prix de la clarté, il serait sans doute possible d'arriver à un modèle cohérent et uniforme.

2.3. Aspect méthodologie

L'intérêt essentiel du système de coopération, outre celui de répondre à un besoin ponctuel, résulte de la démarche pragmatique adoptée.

En effet sur le plan méthodologique, l'introduction d'un outil au niveau de la couche application du modèle de référence répond mieux à l'impératif de perturber le moins possible le système standard du constructeur. En plus, nous nous sommes efforcés, compte tenu des contraintes inhérentes à ce genre d'applications, de trouver un modèle et une méthode qui nous affranchissent des spécificités des systèmes particuliers.

Cependant quels qu'en soient les attraits, ce genre d'approche recueille actuellement peu d'adeptes ; ce pour plusieurs raisons :

- c'est une approche qui est par essence transitoire en attendant qu'elle soit entièrement intégrée au logiciel constructeur dans le cadre d'une normalisation uniformisant les différents niveaux ;
- la faiblesse des débits de transmission des réseaux pénalisant tout transfert de gros volumes de données.

Néanmoins il ne faudrait pas que ce type d'applications soit condamné car avec les nouvelles techniques de transmission à forts débits (notamment par fibres optiques), les contraintes relatives aux délais de transmission peuvent être en partie levées.

2.4. Conditions pour l'intégration de la coopération

Pour mener à bien une coopération surtout en milieu hétérogène, il est nécessaire d'élucider les principaux problèmes ayant trait au contexte général de l'architecture du système réparti. Ces problèmes se posent essentiellement au niveau :

- de la compatibilité des différentes architectures locales,
- de l'identification des fichiers,

- de la compatibilité - des différentes représentations des données et des méthodes d'accès associées,

- des mécanismes de transfert mis en jeu.

1) Compatibilité des architectures

Il y a une nécessité de définir une architecture globale cohérente et uniforme des systèmes informatiques ouverts en environnement réparti.

2) Identification des fichiers

Les techniques d'identification couramment employées actuellement utilisent des catalogues (avec ou sans chemin d'accès). L'identification est en général à deux niveaux et comporte des conventions locales à un système dont on connaît le nom et l'adresse réseau.

Ces techniques, obligeant à changer le nom du fichier à chaque changement de localisation, réclament au moins une gestion transparente de cet espace d'adressage hiérarchisé.

3) Représentation des données

La compatibilité entre différentes représentations de données passe nécessairement par le choix de quelques organisations de données standards qui puissent être transcodées dans chaque système grâce à un langage commun de description de structures.

4) Mécanismes de transfert

La définition de protocoles de transfert de fichiers ainsi que de méthodes d'accès réseau est devenue une nécessité impérieuse.

Ce sont là quelques unes des mesures qu'il nous semble essentiel de mettre en place pour arriver à des systèmes véritablements ouverts.

BIBLIOGRAPHIE

SOMMAIRE BIBLIOGRAPHIQUE

<u>RES</u>	RESEAUX D'ORDINATEURS
<u>SGF</u>	SYSTEMES DE GESTION DE FICHIERS
<u>GFR</u>	SYSTEMES DE GESTION DE FICHIERS REPARTIS
<u>BDR</u>	SYSTEMES DE GESTION DE BASES DE DONNEES REPARTIS
<u>CAC</u>	CONTROLE D'ACCES CONCURRENTS
<u>MAS</u>	MODELES D'ARCHITECTURE DE SYSTEMES
<u>MER</u>	MECANISMES D'EXECUTION REPARTIE
<u>GEN</u>	OUVRAGES GENERAUX

RES : RESEAUX D'ORDINATEURS

- RES-1 J. ANGELIDES
Intégration de systèmes dans un réseau local d'ordinateurs. Application au Centre de Calcul Réparti.
Thèse 3^e cycle, Systèmes et Réseaux Informatiques, Saint-Etienne, Ecole des Mines, Janvier 1980.
- RES-2 BARAN
On distributed Communications.
Rand Corporation, Août 1964.
- RES-3 R. BEAUFILS, J.P. CABANEL, J.P. LAGASSE
ARAMIS: réseau de mini-ordinateurs.
Toulouse, Centre d'Informatique, Univ. Paul-Sabatier, 1975.
- RES-4 M. CART
Développement d'un logiciel de Frontal pour un Centre de Calcul Réparti
Rapport de DEA, Saint-Etienne, Ecole des Mines, Octobre 1977.
- RES-5 M. CART
Fonctions frontales dans un environnement réparti bâti sur un réseau local.
Thèse Doc. ING., Saint-Etienne, Ecole des Mines, Décembre 1979.
- RES-6 J.F. CHAMBON, B. LE BIHAN
Architecture d'un frontal en environnement télé-informatique (Application au réseau CYCLADES).
Thèse Doc. ING., Saint-Etienne, Ecole des Mines, Octobre 1976.
- RES-7 J.F. CHAMBON, S. GUIBOUD-RIBAUD, B. LE BIHAN, Y. TOSAN
Intérêt et faisabilité d'un centre de calcul fondé sur un réseau de mini-ordinateurs.
Saint-Etienne, Ecole des Mines, Octobre 1976.
- RES-8 J.F. CHAMBON, M. JUREMA, A. NIANG
Architecture de fichiers distribués dans un réseau local hétérogène.
ACM-AFCET, Colloque Bases de Données, Tunis, pp. 87-100, Avril 1981.
- RES-9 J.F. CHAMBON, M. JUREMA, A. NIANG
Banalisation de fichiers dans le cadre d'un Centre de Calcul Réparti.
CITEL-AFCET, Congrès Conception des Systèmes Télématicques, Nice, Juin 1981.
- RES-10 M. CHETRIT, A. NIANG
Développement d'un Système Centralisé de Poste Restante sur le réseau CYCLADES : AGORA
Rapport de diplôme ENSEEIHT et de DEA, Toulouse, Juin 1978.
- RES-11 G.E. CONANT, S. WECKER
DNA : an architecture for heterogenous computer networks.
ICCC, Toronto, Août 1976, pp. 618-625.

- RES-12 F.P. CORR, D.H. NEAL
SNA and emerging international standards.
IBM Systems Journal, vol. 8, n° 2, pp. 244-262, 1979.
- RES-13 DECNET
Digital Data Communications Message Protocol DDCMP Specification.
Digital Equip. Corp., Version 4.0, Mars 1978.
- RES-14 DECNET
Digital Network Architecture. Network Services Protocol (NSP).
Digital Equip. Corp., Functional Specification version 3.1. Mars 1978.
- RES-15 H.S. ELOVITZ, C.L. HEITMEYER
What is a computer network ?
IEEE COMPCON FALL, Septembre 1977, pp. 5-3 à 5-10.
- RES-16 D. J. FARBER
A ring network.
Datamation, Février 1975, pp. 44-46.
- RES-17 D. J. FARBER, HOPWOOD, ROWE
Software methods for achieving fail-soft behavior in the distributed computing system.
IEEE Symposium on Computer Software Reliability, pp. 7-11, Avril 1973.
- RES-18 H. FRANK, I.T. FRISCH
Planning Computer-Communication Networks.
Computer communication network, Abramson & Kuo, Englewood Cliffs, Prentice-Hall, 1973.
- RES-19 G.R. GROSSMAN, A. HINCHLEY, C.A. SUNSHINE
Issues in international public data networking.
Computer Networks, vol. 3, n° 4, pp. 259-266, Septembre 1979.
- RES-20 J. HEBENSTREIT
Projet M2 : Conception et réalisation d'un système Multi-Mini-Ordinateurs.
Congrès AFCET, Gif-sur-Yvette, Novembre 1976, pp. 725-732.
- RES-21 M. JUREMA
Gestion du catalogue des utilisateurs dans un Centre de Calcul Réparti.
Rapport de DEA, Saint-Etienne, Ecole des Mines, Octobre 1978.
- RES-22 M. JUREMA, A. NIANG
Système de coopération de systèmes de gestion de fichiers - étude préliminaire.
Saint-Etienne, Ecole des Mines, Novembre 1979.
- RES-23 LORRAINS
Réseaux téléinformatiques
Paris, Hachette, Octobre 1979.

- RES-24 C. MACCHI, J.F. GUILBERT
Téléinformatique: transport et traitement de l'information dans les réseaux et systèmes téléinformatiques.
Paris, DUNOD-Informatique, 1979.
- RES-25 E.G. MANNING, R.W. PEEBLES
A homogeneous network for data-sharing communications.
Computer Networks, vol. 1, N° 4, pp. 211-224, Mai 1977.
- RES-26 J.H. MCFADYEN
Systems Network Architecture : an overview.
IBM Systems Journal, vol. 15, n° 1, pp. 4-23, 1976.
- RES-27 R.M. METCALFE, D.R. BOGGS
ETHERNET: distributed packet switching for local computer networks.
CACM, vol. 19, n° 7, pp. 395-404, Juillet 1976.
- RES-28 S. MIRANDA
Les réseaux d'ordinateurs (étude synthétique)
Toulouse, ENSEEIHT, 1975.
- RES-29 F. PONCET, P. ALLAIRE
Message exchange Protocol for Transpac and Euronet Network Management
ACM-IFIP, Proc. of the Computer Network Protocols, Univ. Liège, Février 1978, pp. C4-1 à C4-8.
- RES-30 R. POPESCU-ZELETIN
The data access and transfer support in a local heterogeneous network (HMINET).
ACM-IEEE, Proc. of the 6th Data Communications Symposium, Californie, Novembre 1979, pp. 147-152.
- RES-31 L. POUZIN
Presentation and major aspects of the CYCLADES computer network.
Proc. of the 3rd Data Communication Symposium, Novembre 1973, pp. 80-87.
- RES-32 L. POUZIN, J.L. GRANGE
Cigale, la machine de commutation de paquets du réseau CYCLADES
Congrès AFCET, Novembre 1973.
- RES-33 L.G. ROBERTS, B.D. WESSLER
The ARPA Network.
ARPA, Washington, D.C., Mai 1971.
- RES-34 A.M. RYBCZYNSKI, J.D. PALFRAMAN
A Common X.25 interface to public data networks.
Computer Networks, vol. 4, n° 3, pp. 97-110, Juin 1980.
- RES-35 S. SEDILLOT
Caractéristiques générales du réseau EIN.
Rocquencourt, INRIA, EIN-501, Mai 1976.

- RES-36 Y. TOSAN
Langages et protocoles dans un Centre de Calcul Réparti.
Thèse 3^e cycle, Systèmes et Réseaux Informatiques, Saint-Etienne, Ecole des Mines, Mars 1978.
- RES-37 TRANSPAC
Manuel Technique TRANSPAC - Spécification Technique d'Utilisation du Réseau.
TRANSPAC - Direction Commerciale, Tour Maine-Montparnasse, Paris, Janvier 1981.
- RES-38 W. WULF, R. LEVIN
A local network.
Datamation, Février 1975, pp. 47-50.
- RES-39 H. ZIMMERMANN
The Cyclades experience : results and impacts.
Congrès IFIP 77, Toronto, Août 1977.
- RES-40 H. ZIMMERMANN
Réseaux informatiques
Techniques de l'Ingénieur, Informatique, H3500, Mars 1980.

SGF : SYSTEMES DE GESTION DE FICHIERS

- SGF-1 CII-HB
Mini-6 - GCOS 6 MOD 600 - Systems Concepts.
CII-HB, 1979.
- SGF-2 DEC
Introduction to VAX-11 Record Management Service.
Digital Equip. Corp., Février 1978.
- SGF-3 DEC
VAX-11 Record Management Service Reference Manual
Digital Equip. Corp., Février 1978.
- SGF-4 M. GALINIER
Introduction aux Fichiers
Toulouse, ENSEEIHT, 1976.
- SGF-5 G. GARDARIN, R. GOMEZ
Gestion des fichiers
Techniques de l'Ingénieur, Informatique, H3120, 1979.
- SGF-6 C. JOUFFROY, C. LETANG
Les Fichiers
Pratique et choix de l'organisation.
Paris, DUNOD-Informatique, 1979.

SGF-7 C. MANOU, C. TRULLEN
Systèmes d'exploitation - Deuxième Partie - Gestion de Fichiers.
Toulouse, ENICA, 1972.

SGF-8 SEMS
Manuel de référence S.T. 1330 - FMS.
SEMS, Septembre 1976.

GFR : SYSTEMES DE GESTION DE FICHIERS REPARTIS

GFR-1 H. BREITWIESER
The management of the Catalog for logical files.
INRIA, Seminar on Distributed Data Sharing Systems, Aix-en-Provence,
Mai 1979, pp. 261-274.

GFR-2 J. C. CHUPIN, J. SEGUIN
MADRE (Méthode d'Accès Direct Réseau).
Grenoble, C.I.I., Avril 1974.

GFR-3 DECNET
Digital Network Architecture. Data Access Protocol DAP.
Functional Specification (version 4.1.).
Digital Equip. Corp., Mars 1978.

GFR-4 D.J. FARBER, F. R. HEINRICH
The structure of a distributed computer system - The distributed file
system.
Proc. of the Int. Conf. on Computer Communications, Octobre 1972, pp.
364-370.

GFR-5 D. K. GIFFORD
Violet, an experimental Decentralized System.
Palo Alto, Xerox, Septembre 1979.

GFR-6 E. HOLLER, H. BREITWIESER, O. DROBNIK , C. KEIL, U. KERSTEN
DISCO : a distributed data base system based on logical files.
INRIA, Seminar on Distributed Data Sharing Systems, Aix-en-Provence,
Mai 1979, pp. 23-39.

GFR-7 J. E. ISRAEL, J. G. MITCHELL, H. E. STURGIS
Separating Data From Function in a Distributed File System.
Palo Alto, Xerox, Septembre 1978.

GFR-8 K. LUNN, K.H. BENNETT
A Highly Reliable Distributed Filestore Directory System.
IEEE - INRIA, Proc. of the 2nd Int. Conf. on Distributed Computing
System, Paris, Avril 1981, pp. 299-307.

- GFR-9 D. SWINEHART, G. M. DANIEL, D. BOGGS
WFS : a Simple Shared File System for a Distributed Environment.
Palo Alto, Xerox, Octobre 1979.

BDR : SYSTEMES DE GESTION DE BASES DE DONNEES REPARTIS

- BDR-1 M. ADIBA
Présentation générale d'un système de coopération de bases de données réparties.
AFCET-IP, Journées de formation: Bases de Données Réparties, Mars 1977, pp. 63-87.
- BDR-2 M. ADIBA, E. ANDRE, J. C. CHUPIN, P. DECITRE, C. DELOBEL, M. LEONARD, GT. NGUYEN, PORTAL, F. REYNAUD, J. RICHY, J. SEGUIN, G. SERGEANT, A. STIERS
POLYPHEME - Propositions pour un modèle de répartition et coopération de Bases de Données dans un réseau d'ordinateurs.
Rocquencourt, INRIA, Bases de Données Réparties, Lans en Vercors, pp. 9-127, Mars 1976.
- BDR-3 P. BOSC
Projet FRERES - Interrogation de Fichiers Répartis sur un Réseau de calculateurs hétérogènes.
Journées BIGRE, Nancy, Janvier 1979.
- BDR-4 P. BOSC, A. CHAUFFAUT
Contribution à l'étude de données réparties
Architecture fonctionnelle du système FRERES.
Thèse Doc. ING., IRISA, Rennes, 1978.
- BDR-5 C. BRUNET, B. LEPAPE
Coopération entre SGBD
AFCET-IP, Journées de formation: Bases de Données Réparties, Mars 1977, pp. 45-62.
- BDR-6 R. DEMOLOMBE, M. LEMAITRE
Rôles d'un modèle commun dans la conception et l'utilisation d'un SGBD réparti: analyse des principaux modèles.
AFCET-IP, Journées de formation: Bases de Données Réparties, Mars 1977, pp. 21-44.
- BDR-7 R. DEMOLOMBE, M. LEMAITRE, J.M. NICOLAS
Pré-étude Projet Pilote : "Système de Gestion de Bases de Données Réparties."
Rocquencourt, INRIA, MOD-I-002, Projet SIRIUS, Mars 1976.
- BDR-8 C. ESCULIER, L. TREILLE
SIRIUS-DELTA : un prototype industriel réalisé à l'IRIA en collaboration avec INTERTECHNIQUE.
INRIA, Bulletin de Liaison n° 57, Octobre 1979.

- BDR-9 V. FAUSER, E.J. NEUHOLD
The Architecture of distributed Relational DBMS POREL
INRIA, Seminar on Distributed Data Sharing Systems, Aix-en Provence Mai 1979.
- BDR-10 G. GARDARIN
Présentation du Système de Gestion de Bases de Données Réparties SIRIUS-IR.
Rocquencourt, INRIA, SCH-I-014, Projet SIRIUS, Février 1977.
- BDR-11 G. GARDARIN, M. JOUVE, C. PARENT, S. SPACCAPIETRA
Analyse de la conception d'un système de gestion de bases de données réparties.
AFCET-IP, Journées de formation: Bases de Données Réparties, Mars 1977, pp. 89-111.
- BDR-12 J. LE BIHAN
La répartition des données dans les Réseaux Informatiques
Congrès AFCET, Gif-sur-Yvette, Novembre 1976, pp. 55-65.
- BDR-13 J. LE BIHAN, C. ESCULIER, G. LE LANN, L. TREILLE
SIRIUS-DELTA : Un prototype de Système de Gestion de Bases de Données Réparties.
INRIA, Proc. Int. Symp. on Distributed Data Bases, Paris, Mars 1980.
- BDR-14 D. MEYER, M. DESFEUILLET
Comment passer d'une base de données locale à une base de données répartie ? Exemple de la gestion comptable et financière du CNRS.
AFCET-IP, Journées de formation: Bases de Données Réparties, Mars 1977, pp. 209-230.
- BDR-15 J.B. ROTHNIE
SDD-1 : A System for Distributed Databases.
Computer Corporation of America, Technical Report, Juillet 1978.
- BDR-16 S. SPACCAPIETRA
Problématique de conception d'un système de gestion de base de données réparties.
Thèse d'Etat, Univ. PARIS VI, Novembre 1978.
- BDR-17 M. STONEBRAKER, E. NEUHOLD
A distributed database version of INGRES
Proc. Berkeley Workshop on Distributed Data Management and Computer Networks.
Berkeley, Mai 1977.
- BDR-18 R.A.C. THOMAS
Process structure alternatives towards a distributed INGRES
INRIA, Proc. Int. Symp. on Distributed Databases, Paris, Mars 1980.
- BDR-19 K.C. TOTH, S.A. MAHMOUD, J.S. RIORDON, O. SHERIF
The ADD System : An Architecture for Distributed Databases.
Proc. of the 4th Int. Conference on VLDB, Septembre 1978.

CAC : CONTROLE D'ACCES CONCURRENTS

- CAC-1 D.Z. BADAL
 On the degree of concurrency provided by concurrency control mechanisms for distributed databases.
 Proc. of the Int. Symposium on Distributed Data Bases, Paris, Mars 1980, org. INRIA, Delobel et Litwin ed. Amst. N.H., pp. 35-48.
- CAC-2 P. A. BERNSTEIN, J. B. ROTHNIE, N. GOODMAN, C. A. PAPADIMITRIOU
 The concurrency control mechanism of SDD-1: a system for distributed databases (The fully redundant case).
IEEE Trans. on Software Engineering, Vol. SE-4, n° 3, Mai 1978, pp. 154-168.
- CAC-3 C. A. ELLIS
 A robust algorithm for updating duplicate databases.
 Proc. of the 2nd Berkeley Conf. on Distributed Data Management and Networks, Mai 1977, pp. 146-158.
- CAC-4 K.P. ESWARAN, J.N. GRAY, R.A. LORIE, I.L. TRAIGER
 The notions of consistency and predicate locks in a database system.
CACM, Vol. 19, N° 11, Novembre 1976, pp. 624-633.
- CAC-5 G. GARDARIN
 Problems and solutions for updating multiple and distributed copies of data.
 Rocquencourt, INRIA, CTR-I-007, Projet SIRIUS, 1979.
- CAC-6 M. JUREMA
 Contrôle de concurrence d'accès à des données dans les systèmes répartis.
 Saint-Etienne, Ecole des Mines, Novembre 1980.
- CAC-7 G. LE LANN
 An overview of distributed control techniques.
 Advanced course on "The design of distributed processing systems", EEC-INRIA, CTR-I-012, Juin 1978.
- CAC-8 G. LE LANN
 Algorithms for distributed data-sharing systems which use tickets.
 Proc. of the 3rd Berkeley Workshop on Distributed Data Management and Computer Networks, Août 1978, pp. 259-272.
- CAC-9 G. LE LANN
 Le contrôle dans les systèmes informatiques répartis: nature du problème et quelques solutions.
Journées BIGRE, Nancy, Janvier 1979, INRIA, CTR-I-005, pp. 266-286.
- CAC-10 G. LE LANN
 Présentation des activités du groupe SCORE.
 Rocquencourt, INRIA, ORG-I-037, Projet SIRIUS, Mars 1979.

- CAC-11 D. J. ROSENKRANTZ, R. E. STEARNS, P. M. LEWIS II
System level concurrency control for distributed database systems.
ACM Trans. on Database Systems, Vol. 3, N° 2, pp. 178-198, Juin 1978.
- CAC-12 F. B. SCHNEIDER
Ensuring consistency in a distributed database system by use of distributed semaphores.
Proc. of the Int. Symposium on Distributed Data Bases, Paris, Mars 1980, org. INRIA, Delobel et Litwin ed. Amst. N.H., pp. 183-189.
- CAC-13 W. SEEMANN
Concurrency control in the distributed file management system DISCO.
INRIA, Seminar on Distributed Data Sharing Systems, Aix-en-Provence, Mai 1979, pp. 293-306.
- CAC-14 J. SEGUIN, G. SERGEANT, P. WILMS
Cohérence et gestion d'objets dupliqués dans les systèmes distribués.
Grenoble, Rapport de Recherche CICG/ENSIMAG, 1977.
- CAC-15 R. H. THOMAS
A majority consensus approach to concurrency control for multiple copy databases.
ACM Trans. on Database Systems, Vol. 4, N° 2, Juin 1979, pp. 180-209.

MER : MECANISMES D'EXECUTION REPARTIE

- MER-1 R.G. CASEY
Allocation of copies of a file in an information network.
AFIPS, SJCC, vol. 40, Mai 1972, pp. 617-625.
- MER-2 CORNAFION
Désignation dans les réseaux d'ordinateurs
Document n° 3 (version 1), Décembre 1977.
- MER-3 P.R.F. CUNHA, T.S.E. MAIBAUM
A communication data type for message oriented programming.
Proc. of the 4th Int. Symposium on Programming, Paris, Avril 1980, ed. B. ROBINET, pp. 79-91.
- MER-4 NG.X. DANG, J.M. DA GRACA MARTINS, G. SERGEANT
SIGOR: Système d'Interprétation Généralisé Orienté Réseau
Un outil pour l'écriture et la mise au point d'applications réparties
Journées BIGRE, Nancy, Janvier 1979.
- MER-5 D.J. FARBER, K.C. LARSON
The system architecture of distributed computer system - The communication system.
Proc. of the Symposium on Computer Communication Networks and Teletraffic, Avril 1972, pp. 21-27.

- MER-6 M.J. FISCHER, L.J. GUIBAS, N.D. GRIFFETH, N.A. LYNCH
Optimal placement of identical resources in distributed network.
IEEE - INRIA, Proc. of the 2nd Int. Conf. on Distributed Computing Systems, Paris, Avril 1981, pp. 324-336.
- MER-7 D. HERMAN
Contrôle Réparti des synchronisations entre processus
INRIA, 2nd Int. Conf. on Distributed Computing Systems, Versailles, Avril 1981.
- MER-8 E. HOLLER
Allocation of distributed resources.
INRIA, Seminar on Distributed Data Sharing Systems, Aix-en-Provence, Mai 1979.
- MER-9 U. KERSTEN
Access management for logical files and recovery procedure in the distributed file management system DISCO.
INRIA, Seminar on Distributed Data Sharing Systems, Aix-en-Provence, Mai 1979, pp. 275-291.
- MER-10 G. LE LANN, R. NEGARET
Une approche unifiée des systèmes informatiques répartis à travers le concept de communication
Congrès AFCET, Gif-sur-Yvette, pp. 561-572, Novembre 1976.
- MER-11 B. LINDSAY
Object naming and catalog management for a distributed database manager.
IEEE - INRIA, Proc. of the 2nd Int. Conf. on Distributed Computing System, Paris, Avril 1981, pp. 299-307.
- MER-12 G. LOUIT
Les protocoles de ARPANET
Rocquencourt, Colloque IRIA/ACM, Réseaux d'ordinateurs, Mars 1972.
- MER-13 S. MAHMOUD, J.S. RIORDON
Optimal allocation of ressources in distributed information networks.
ACM Trans. on Database Systems, Vol. 1, N° 1, Mars 1976.
- MER-14 R. NEGARET
Etude de l'allocation de ressources dans les systèmes informatiques répartis.
Thèse 3^{ème} Cycle, Informatique, Univ. de Rennes, Décembre 1976.
- MER-15 G. SERGEANT, M. FARZA
Machine interprétative pour la mise en oeuvre d' un langage de commande sur le réseau CYCLADES.
Thèse 3^{ème} Cycle, Univ. Paul Sabatier, Toulouse, Octobre 1974.
- MER-16 H. ZIMMERMANN, J.S. BANINO, A. CARISTAN, M GUILLEMONT, G. MORISSET
Basic Concepts for the support of distributed systems : the CHORUS approach
INRIA, 2nd Int. Conf. on Distributed Computing Systems, Versailles, Avril 1981.

MAS : MODELES D'ARCHITECTURE DE SYSTEMES

- MAS-1 G.A. ANDERSON, E.D. JENSEN
Computer interconnection structures: taxonomy, characteristics and
exemples.
Computing Surveys, vol. 7, n° 4, pp. 197-213, Décembre 1975.
- MAS-2 ANSI/X3/SPARC Study Group on Data base Management Systems
Interim Report
Bulletin of the ACM SIGMOD, Vol. 7 n° 2, Février 1975.
- MAS-3 C. BACHMAN
Distributed Systems - Requirements and Architectural Considérations.
INRIA, International Symposium on Distributed Data Bases, Invited
Speaker, Paris, Mai 1980.
- MAS-4 D.R. BOGGS, J.F. SHOCK, E.A. TAFT, R.M. METCALFE
Pup: an internetwork architecture.
Palo Alto, Xerox, CSL-79-10, Juillet 1979.
- MAS-5 E.F. CODD
A Relational Model of Data for Large Shared Data Banks
CACM, Vol. 13, n° 6, Juin 1970.
- MAS-6 J. DONNELLEY
Components of a Network Operating System.
Computer Networks, Vol. 3 , n° 6, pp. 389-399, Décembre 1979.
- MAS-7 G. GARDARIN
A unified Architecture for Data and Message Management.
Proc. AFIPS, Vol. 48, pp. 681-688, Juin 1979.
- MAS-8 G. GARDARIN, Z. BELLAHSENE, P. FAUDEMAY, R GOMEZ, K. KARLSSON,
M. THONNET, P. VALDURIEZ
Objectifs, principes et architecture d'une Machine Bases de Données
Réparties.
ACM-AFCET, Colloque Bases de Données, Tunis, pp. 317-355, Avril 1981.
- MAS-9 M. GIEN
A File Transfert Protocol (FTP).
Computer Networks, vol. 2, n° 4-5, pp. 312-319, 1978.
- MAS-10 M. GIEN, H. ZIMMERMANN
Design principles for Network Interconnection.
ACM-IEEE, Proc. 6th Data Com. Symposium, Novembre 1979, pp. 109-119.
- MAS-11 S. GUIBOUD-RIBAUD
Architecture des systèmes
Une vue synthétique : logicielle et matérielle.
Saint-Etienne, Ecole des Mines, 1978.
- MAS-12 ISO/TC97/SC16/N227
Reference model of open system interconnection.
Août 1979.

GEN : OUVRAGES GENERAUX

- GEN-1 A.T. BERZTISS.
Data structures - theory and practice.
Academic Press, Inc., 1973.
- GEN-2 C. BETOURNE
Systèmes d'exploitation
Toulouse, ENSEEIHT, 1977
- GEN-3 DEC
VAX11/780 SOFTWARE HANDBOOK Vol. 3.
Digital Equip. Corp., 1977.
- GEN-4 J.M. FORESTIER
Machine relationnelle pour système segmenté.
Thèse 3° cycle, Génie Informatique, Univ. Grenoble, Septembre 1978.
- GEN-5 C.C. GOTLIEB, L. R. GOTLIEB.
Data Types and Structures.
Englewood Cliffs, Prentice-Hall, 1978.
- GEN-6 R. MAHL, J.C. BOUSSARD.
Algorithmique et structures de données.
Univ. de Nice, Laboratoire d'Informatique, 1977.

AUTORISATION DE SOUTENANCE

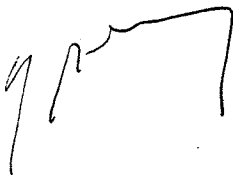
VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,
VU les rapports de présentation de M. J.F. CHAMBON et J.C. CHUPIN

MM. JUREMA et NIANG


sont ~~est~~ autorisés à présenter une thèse en soutenance pour l'obtention
du diplôme de DOCTEUR-INGENIEUR, spécialité Systèmes et Réseaux Informatiques

Fait à Saint-Etienne, le 25 juin 1981

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble



Le Directeur de l'EMSE,



M. MERMET
INGENIEUR EN CHEF DES MINES
DIRECTEUR
ECOLE NATIONALE SUPERIEURE DES
MINES DE SAINT-ETIENNE

